Middlesex University London

Middlesex University Research Repository

An open access repository of

Middlesex University research

http://eprints.mdx.ac.uk

Surdhar, Jagpal Singh (1999) Fuzzy PD control of an optically guided long reach robot. PhD thesis, Middlesex University.

Accepted Version

Available from Middlesex University's Research Repository at http://eprints.mdx.ac.uk/13503/

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this thesis/research project are retained by the author and/or other copyright owners. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge. Any use of the thesis/research project for private study or research must be properly acknowledged with reference to the work's full bibliographic details.

This thesis/research project may not be reproduced in any format or medium, or extensive quotations taken from it, or its content changed in any way, without first obtaining permission in writing from the copyright holder(s).

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

Middlesex University London

Middlesex University Research Repository:

an open access repository of
Middlesex University research

http://eprints.mdx.ac.uk

Surdhar, Jagpal Singh, 1999.
Fuzzy PD control of an optically guided long reach robot.
Available from Middlesex University's Research Repository.

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this thesis/research project are retained by the author and/or other copyright owners. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge. Any use of the thesis/research project for private study or research must be properly acknowledged with reference to the work's full bibliographic details.

This thesis/research project may not be reproduced in any format or medium, or extensive quotations taken from it, or its content changed in any way, without first obtaining permission in writing from the copyright holder(s).

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:
eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

Fuzzy PD Control of an Optically Guided Long Reach Robot

Thesis submitted to Middlesex University in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Jagpal Singh Surdhar

School of Engineering Systems
Middlesex University, London (UK)

April 1999

For you, Mother, who gave me the chance to study.

Abstract

This thesis describes the investigation and development of a fuzzy controller for a manipulator with a single flexible link. The novelty of this research is due to the fact that the controller devised is suitable for flexible link manipulators with a round cross section. Previous research has concentrated on control of flexible slender structures that are relatively easier to model as the vibration effects of torsion can be ignored. Further novelty arises due to the fact that this is the first instance of the application of fuzzy control in the optical Tip Feedback Sensor (TFS) based configuration.

A design methodology has been investigated to develop a fuzzy controller suitable for application in a safety critical environment such as the nuclear industry. This methodology provides justification for all the parameters of the fuzzy controller including membership functions, inference and defuzzification techniques and the operators used in the algorithm. Using the novel modified phase plane method investigated in this thesis, it is shown that the derivation of complete, consistent and non-interactive rules can be achieved. This methodology was successfully applied to the derivation of fuzzy rules even when the arm was subjected to different payloads. The design approach, that targeted real-time embedded control applications from the outset, results in a controller implementation that is suitable for cheaper CPU constrained and memory challenged embedded processors.

The controller comprises of a fuzzy supervisor that is used to alter the derivative term of a linear classical Proportional + Derivative (PD) controller. The derivative term is updated in relation to the measured tip error and its derivative obtained through the TFS based configuration. It is shown that by adding 'intelligence' to the control loop in this way, the performance envelope of the classical controller can be enhanced. A 128% increase in payload, 73.5% faster settling time and a reduction of steady state of over 50% is achieved using fuzzy control over its classical counterpart.

Contents

Acknowledgements

I wish to express my sincerest gratitude to the following:-.

Professor Tony White, a director of studies that I have been proud to have. Tony has been an invaluable source of knowledge and direction during the course of this project. Completion of this project would certainly have not been possible without his advice, critical assessment and friendly encouragement.

Dr Raj Gill and Peter Warner, my supervisors for this research. Raj was instrumental in the smooth running of the project including all the administrative and money matters and Peter was always at hand to offer intelligent and useful insight to the research. I would also like to thank Dr. Mark Stoker, in the capacity of research advisor, for his invaluable help on the intricacies of transputers.

J.A. Schofield and Gulab Mistry, from British Nuclear Fuels (BNF) plc. UK, whose continued interest kept this project in focus.

Issam Siman and Imtiaz Bhaijii, the robotics laboratory technicians, who would drop everything to help. Their friendly banter and twisted humour took the tedium out of some long and hot summer days being locked in a room with a robot.

My fellow research colleagues, now my dear friends, Dr. J. V. Miro (Jaime), Bernard Parsons and Dr. J. B. Lewis (Jeremy). Jaime for the many hours spent over coffee and pint discussing topics ranging from non-linear control to payella, and robots to scuba diving. I am also indebted to Jaime and Emma for their kindness in letting me 'kip on the floor' whenever I came down to London to work on my project. Bernard, for his lectures on C++, being a task master in the gym, and devising sprinting and trolley pushing exercises. Jeremy, 'I can build anything from scrap including the TFS based robot', whom I had the pleasure of sharing an office with, for being ever willing to read drafts of thesis and for the inspiration I derived from his inventiveness. Thanks too to my other research colleges, John Korhonen, Stefan Okrongli, Ray Chanmugam and A. Lasebe whom I interacted with during the course of the project and who made up the friendly research atmosphere in the Advanced Manufacturing and Mechatronics Centre (AMMC) lab.

My special thanks to the patience of Frida and Irene, the secretaries of the School of Engineering Systems, whom I constantly pestered whilst I floundered my way through administrative procedure. Thanks too to Sandra Gemilli for giving me part-time work in Admissions.

I would also like to give special thanks to Professor T. Takamori, Dr. S. Tadokoro, S. Kobayashi, M. Hattori, R. Kanno and the graduate students of the Department of Computer and Systems Engineering at Kobe University for being excellent hosts during our month long stay on a collaborative research project in Kobe, Japan.

Professor T. J. Tarn of Washington State University for instilling the fear of the right hand side zero dynamic into me whilst we shared a taxi in Vienna.

Professor David Wang of Waterloo University's Department of Electrical and Computer Engineering, Canada, for taking the time out of his busy schedule to discuss, via email and fax, the intricacies of modelling flexible link manipulators with modified output measurements. I would like also to thank the researchers of Waterloo University's Control of Flexible Structures Research Group (ConStruct) for providing me with useful references.

Dr. Peter Bell of Cambridge Consultants Limited, my group leader, for his understanding and giving me the time to complete my thesis.

I wish to thank my supportive family for always being there through thick and thin. Especially Christina, my longsuffering wife, for bearing with me while I was often 'somewhere else' and lost in thought, calming me when I was stressed and putting up with my irritating habits of scribbling

an idea that couldn't wait on MacDonalds or Burger King napkins. Christina also 'manned' fort whilst I was working unsociable hours at the university and had to work long hours, including commuting 3 hours a day, in order to supplement my studentship and part-time teaching to support the family. A special thanks to Alistair and Emma for suffering a grumpy and boring daddy. Thankyou too, mom, Deepi, for showing me never to give up, and dad, Sindi, for showing me patience, dedication and discipline. Thanks too, Bilo, my younger brother, whose hard working nature has been an inspiration. I reserve special mention for Arvindra who has been a never failing big brother and 'mentor' during my academic years.

I would also like to formally thank the following bodies for the funding I received for the project. The AMMC, for my research studentship, and BNF plc. UK, for providing funding for the equipment costs. I am also grateful for travel grants received from BNF plc. UK, The Royal Society (London) and The British Council (Tokyo).

The thesis is based on work undertaken between April 1994 and October 1998. The work has been carried out in the School of Engineering Systems at the Faculty of Technology.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The need for long reach manipulators

The Robotic Industries Association (RIA) definition of a 'robot' is:-

> *A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialised devices through variable programmed motions for the performance of a variety of tasks - RIA [1].*

This definition covers a wide scope of robotic applications including mobile robots capable of autonomous locomotion, remotely controlled surgical instruments and the robotic arms or manipulator arms that are more common to industry [2]. This thesis concentrates on the latter type of robot that are often referred to as 'robotic manipulators'. The field is further narrowed to the study of the control of 'long reach manipulators'. These are, as their name infers, manipulators that have a long reach through having lengthy links.

The main motivation behind the study comes from the remote handling capability of a long reach manipulator to remove human operators from hazardous environments, particularly the radioactive environment in a nuclear power-station.

Potential application areas for a long reach manipulator in the nuclear industry are in decommissioning related activities, for example:-

- In vitrification cells where long reach and dexterity are required,
- and as a deployment boom that could be used to extend the reach of a traditional manipulator.

The suitability of a robot for operation in this environment is subject to the following constraints or specifications:-

- It must be safe to use, implying a stable and robust controller.
- It must have the ability to cope with different payloads, since the robot would be required to carry out a number of different tasks.
- High accuracy and repeatability of end-effector motion are desirable especially when dextrous tasks are to be performed.

A secondary motivation behind this research is to widen the scope of robotic manipulator applications. Most of the commercially available robots used in industry have a small work envelope and a low payload capacity of about 10% of the arm weight [3], having a maximum payload of 5 kg [4][1].

[1]These statistics apply to robots driven by electrical actuators

The more popular application areas for such robots in the United Kingdom include welding, palletising, painting and in-process handling in the pharmaceutical and food industries [5]. Extension of the reach and payload capacity of robot manipulators would allow them to be utilised in the following potential areas:-

- Civil engineering and construction [6][7].
- The nuclear industry [8].
- Space [9].
- Underwater [10].

1.2 Lightweight manipulators

Controllers for the more popular commercially available manipulators work on the assumption that the links are rigid [11]. An 'open loop' control configuration is used. By 'open loop' it is meant that the position of the end-effector is not taken in account although the joints themselves may be controlled in a closed loop fashion. The more obvious way to extend the links of traditional manipulators would be to 'scale-up' the robot i.e., use existing materials, such as steel and aluminium, to build larger and longer links. The links would have to be considerably heavier in order to preserve their rigidity. Extremely high motor torques will be required to maintain the fast manipulation speed of the original version. Large actuators which are powerful enough to produce the necessary torques are very costly and may not even be available. Heavy links result in sluggish, energy inefficient and low accuracy manipulators [12][13].

An alternative to using the existing materials is to maximise the stiffness-to-weight ratio of the arm. This can be done through modifications of structural design and the use of advanced composites [14]. Composites, for example alloys, carbon fibre or graphite/epoxy, are generally lighter and stiffer and result in robot links with a relatively lower inertia in comparison to one fabricated with conventional materials [15][16]. These exotic materials with unique physical properties need to be synthesised in order to meet with the criteria of an optimal design [17]. Since such materials are not obtained 'off-the-shelf' great costs are incurred in terms of design effort and material.

A different approach involves constructing the longer links from existing readily available materials and accepting the inevitable flexibility. The resulting lightweight manipulators have several advantages over their rigid counterparts [18]:-

- Quicker system response.
- Less power consumption.
- Smaller actuators required.
- Less mass and generally lower material and actuator costs.

1.2.1 Problems due to link flexibility

Although the 'hardware' costs can be kept low using lightweight manipulators, there will be an inevitable increase in controller complexity to accommodate for the link flexibility. During robot motion the flexible link may experience vibrations and distortion resulting in poor positioning accuracy and an oscillatory transient response [19]. These vibrations must be dissipated prior to undertaking subsequent manoeuvres such as an assembly operation, and clearly the associated settling-time must be minimised if cycle times are to be reduced. 'Advanced' control techniques, requiring sophisticated dynamic models of the system, are often required to compensate for the effects of bending and torsion in order to achieve fast and accurate end-effector placement without overshoot [20].

1.2.2 Sensing requirements for flexible link robots

When designing controllers for most commercially available robots, it is assumed that the links are inelastic and that no permanent link distortion takes place. 'Conventional' hub-based controllers that utilise sensing other than end-point position sensing cannot guarantee accurate tip positioning [21][22][23].

The most direct approach to the correction of tip position errors for a manipulator would be to use end-point sensing where the tip position of the arm is measured with reference to a known external point. These measurements can then be incorporated in the control scheme to minimise the measured error during motion and the effects of manipulator dynamics, gravitational loading and drive flexibility can be compensated [24].

Usually the same hub-located actuators which cause the initial motion are used to correct for the deflections measured at the arm tip. Such systems which consist of a dynamic system (the flexible arm) separating the actuator and sensor are referred to as *non-collocated* [25].

Non-collocated control methods, especially using direct tip position sensing, have distinct advantages in terms of positioning accuracy. These are:-

- Compensation for gravitational loading.
- Compensation for drive flexibility.
- Compensation for link flexibility.
- Compensation for both permanent and elastic link deformation.

However, because there is a finite propagation time for a bending wave to travel along the flexible arm between the actuator and sensor, there will be problems with controller stability. This lag cannot be ignored in the controller design [26]. This thesis aims to address these the inherent problems of non-collocation through the introduction of a novel controller based about a unique physical configuration. Tip feedback sensor (TFS) based control is described in the following section.

1.3 Tip Feedback Sensor (TFS) based control

> *With end-point sensing, feedback alone (suitably conditioned) is sufficient to whip the tip to the commanded position and hold it there precisely. Even more important, a shift in, for example, workpiece with respect to robot base, no longer produces error - Cannon and Schmitz [27].*

The problem of accurate position control of the end-effector of a manipulator can be divided into two main tasks, i.e., trajectory definition and end-point tracking. The former task involves the definition of a suitable and feasible path that the arm may follow in order to achieve the desired position, this is the inverse kinematics problem. The latter task involves keeping the end-effector on the prescribed path, usually this requires the solution of a complex dynamic equation incorporating link flexibility to obtain the necessary demand motion and force signals.

Investigations in this thesis describe a novel control solution, first devised by Lewis and Scott [28][29], to the problem of end-point position control of a flexible manipulator. The novelty arises from the unique hardware configuration of the manipulator system. The control operation is divided into two subsystems:-

- A *master* optical positioning head for defining end-point position,
- and a *slave* manipulator with an optical end-point sensor which causes the arm to follow the master positioning signal via hub based actuators.

The principle of operation is based about feedback of the position error measured at the sensor located at the tip of the manipulator arm. Herein, this type of control is referred to as *tip feedback sensor (TFS)* based control.

The master positioning head, comprising of a laser mounted on stepper motors that can be moved in both horizontal and vertical planes, is used to provide the demand position for the end-effector of the arm. A sensor, located at the tip of the arm, is used to measure deflection from the demand position in both the horizontal and vertical planes. Errors between the demand and actual position signals are then used in a feedback loop. The arm and actuators form the slave subsystem. Fig. 1.1 depicts the TFS based system.

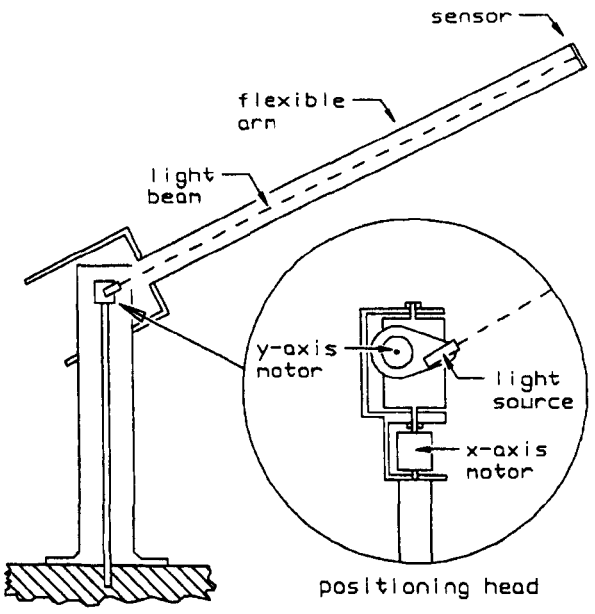

Figure 1.1: Schematic diagram of the 'master/slave' TFS control system [30].

1.3.1 Advantages of TFS based control

The important advantages of TFS based control are :-

- Accurate trajectory tracking and positioning - tip position feedback ensures that the tracking and positioning errors are being minimised during motion and at rest, in addition, there is the further possibility of correction of any tip position steady-state errors at the end of motion as these are directly measurable,
- Simple regulator loop - simple closed loop position feedback about the tip is used for end-point tracking and a solution of a complex dynamic model of the system is not required to generate the necessary demand motion and force signals. The extra bandwidth gained as a result of **not** having to compute a model can be used to achieve faster controller sample rates and to explore Artificial Intelligence (AI) based controllers which can have large processing overheads,
- Integrated trajectory tracking and tip deflection sensing - this is an important advantage for control purposes as only a single control loop is required to accurately position the arm and compensate for link deformations. In contrast, systems which use separate control loops for positioning and deflection compensation are susceptible to controller instability as situations can arise when both loops are competing.
- Sealed laser path command signal - the laser command signal is protected against being cut off by obstacles in its path as it is confined to shine up the hollow arm.
- Fewer sensors - a single tip sensor is used to detect movement in the horizontal and vertical planes for a single link configuration.

1.3.2 Performance of classical controllers in the TFS based configuration

Lewis [30] investigated different classical controllers applied in the TFS based configuration. His experiments were conducted on a prototype robot constructed by him at Middlesex University. The main results of Lewis's investigations on the control of the manipulator using TFS based are summarised below:-

Proportional control Using Proportional control it was possible to control primary arm positioning and correct for arm bending. Proportional control was only suitable for slow actuation speeds - faster actuation led to uncontrollable oscillations at the arm tip.

Other classical control algorithms Proportional plus derivative (PD) and proportional plus derivative plus integral (PID) control were investigated to improve damping and to reduce steady state error. PD control was found to yield superior performance giving robust behaviour with little steady state error. Results for step input and tracking response tests supported these findings. With a step input the settling time at the tip of a 1 m unloaded arm, under PD control, was less than 0.9 s and the steady state error was 0.45 mm. Tracking errors of less than 0.43 mm at the tip were recorded for the arm following both circular and square trajectories.

1.4 Fuzzy control

The research in this thesis expands on the work of Lewis. He showed that it its possible control the end-point position of a long reach flexible robot using the TFS based configuration. His experiments on classical controllers applied in the TFS based configuration have shown that PD control yields the best tracking and transient performance. It is common knowledge that there are limitations to classical control especially when the plant to be controlled is highly non-linear in nature and is subject to non-linear effects such as coulomb friction, flexure in the harmonic drives or gear backlash. Parameters for the classical PD controller of Lewis were tuned under a 'best compromise' criterion in order to give reasonable tracking and step responses. This kind of tuning is far from optimal and there is potential for much improvement in performance should an appropriate controller that adapts to its operating state be used.

In order to comply with the specification laid down in section 1.1, the robot must be able to handle different tip payloads. Different payloads make the robot even more non-linear and the 'best compromise' tuning cannot be easily applied. A controller that is non-linear and adaptive to the dynamics of the robot is required to overcome these limitations [31]. There are many control methods ranging from Linear Quadratic Gaussian (LQG) control [27], Liapunov control [32], optimal switching [33] and adaptive control [31] that would satisfy most of the conditions cited above. The majority of these methods rely on highly accurate mathematical models of the system and sometimes require complex on-line identification especially when non-linearity and changing payloads prevail.

One of the advantages and attractive features of TFS based control is that it *does not rely on an accurate dynamic model of the system for control.* The physical set up of the system already gives a reliable tip position output regarding the command set by the positioning head. The control task is simply to minimise this measured error. This is the basis of feedback control [34]. In such a case, where solution of a mathematical model is not required to achieve the desired end-effector positioning, methods based on the Artificial Intelligence (AI) [35][36][37][38] approach may be used to improve control performance. Fuzzy control is investigated in this thesis as a means of making the TFS based controller adaptive.

The following section defines explicitly the aims of the research by describing the problem, the constraints in solving it and how the performance of the new system will be evaluated.

1.5 Problem Definition

The problem is to devise an adaptive controller for the TFS based prototype robot developed at Middlesex University. The TFS based robot can be categorised as an direct end-point controlled robot. This type of robot is susceptible to problems that arise due to non-collocation of the sensor and actuator. The control problem is further complicated by the fact that the manipulator to be controlled has a round cross section. The round cross section can give rise to additional infinite torsion vibration modes that are not easily modelled.

Previous research has tended to focus on manipulator arms that have slender structures that are relatively easily modelled since the effects of torsion are not included. A further complication is due to the fact that dual axis control of manipulator is required. The implication of this is that the controller must be able to cope with the effects of gravity. The majority of research into the development of flexible link robots has been with space applications in mind and the effects of gravity were ignored in the experimental set-up. Controllers developed using the simpler models would not be suitable for controlling the TFS based robot. As well as having a round cross section, and the controller requirement to cope with the effects of gravity, the Middlesex robot is a prototype to prove the TFS based scheme and suffers from mechanical imperfections that make it even more difficult to model. This contrasts with earlier experiments on flexible manipulator control that have relied on well behaved structures that can be modelled more easily.

A further requirement of the controller being developed is that it must be capable of being deployed in a safety critical environment, particularly the nuclear industry. This calls for a controller that must prove to be stable and robust to external disturbances.

1.5.1 Constraints

There are three main constraints to solving the control problem defined above. The first of these is that the research of the controller is restricted to experimentation on the TFS based prototype robot developed at Middlesex University. This constraint arises as a requirement from the sponsors of this research, BNF plc., who sponsored the research and have funded the purchase of actuators for the robot .

The second constraint is that the research is confined to investigation of AI based methods of control that do not require mathematical models of the system. This constraint is in line with taking advantage of the simplicity of the TFS based configuration. Although there are other non-model based approaches to control, such as self tuning control, this research is constrained to investigation of a fuzzy logic based controller.

Simultaneous real-time and dual axis control with the possibility of extension of control to multiple axes requires a controller that is easily scaleable. This requires the control algorithm to be light, in terms of memory requirements and processing overheads, especially if the controller is to be implemented on cheaper embedded microprocessors rather than on a PC. This is the third constraint.

1.5.2 Assessment of the controller

The following specifications have been laid down to assess the performance of the controller:-

- Payload capacity of up to 5 times the unloaded manipulator arm.
- Steady state error of less than 0.15 mm in response to a step input.
- Peak overshoot of 5% for the highest payload in response to a step input.
- The controller must be stable and robust so it can be applied in a safety critical environment.
- The controller must be able to support dual axis control i.e., it must be able to cope with the effects of gravity on the manipulator.

In addition to the above specifications the controller must be able to outperform classical control which has been previously applied to control the robot. The classical controller of Lewis [30], as described in section 1.3.2, will provide a direct benchmark for controller comparison.

1.6 Aims and objectives of the research

This thesis aims to demonstrate that control based on an AI method, specifically, a supervising fuzzy PD controller based on the TFS configuration, satisfies the specification of a stable, model-free, robust, real-time non-linear adaptive control system. The approach to meeting the research objectives described above is outlined below:-

- Review the published literature describing previous relevant contributions.
- Development and validation of a non-linear model of the prototype robot using Advanced Continuous Simulation Language (ACSL). This model is to be used mainly for comparative performance evaluation of different controllers and as a design guide for the fuzzy controller.
- Investigate issues related to the real time implementation of fuzzy controllers.
- Development and simulation of a real-time intelligent adaptive non-linear controller in the TFS based configuration.
- Implementation of the intelligent controller in an embedded environment.
- Controller performance testing on the real system.
- Comparison and analysis of the results.
- Recommendations for further work.

1.6.1 Statement of originality

The material in this thesis is the original work of the author. Contributions made by undergraduate students were under the supervision of the author, who conceived and proposed the project concepts.

1.7 Outline

The general outline of the contents of this dissertation are as follows:-

Chapter 2 reviews practical implementations of controllers based on end-point or tip sensing. The contribution in this chapter is the presentation of the literature review in a concise tabular form that can be easily referred to by other researchers.

Chapter 3 develops a model of the prototype robot. The investigations include identification of the sources of non-linearity and vibrations in the system using frequency analysis and transient response tests. Conditions for the stability of classical control in the TFS configuration are also investigated. There are no significant contributions in this chapter apart from showing the simulated TFS based system is stable under classical control, however, it has been included for completeness as it gives greater insight into the behaviour of the robot.

Chapter 4 covers the study of how fuzzy control may be applied to the TFS based robot. This chapter investigates the suitability of fuzzy control, reviews relevant fuzzy set theory and related research and gives design recommendations of how fuzzy control can be applied to meet the specification of safe real-time adaptive non-linear control. The main contributions in this chapter are collation of the observations of other workers that have worked on the TFS based robot, and the design recommendations for a fuzzy controller that can be implemented in real-time on embedded processors in a safety critical environment.

Chapter 5 describes the methodology for the design of the variable payload fuzzy PD controller. Part of the design includes justification of the choice of controller variables and fuzzy rules. The contributions of this chapter are the collection and analysis of the robot data and derivation of appropriate fuzzy rules for the controller using the modified phase plane method.

Chapter 6 describes the implementation of the controller in real-time on a network of transputers. Implementation issues such as the resolution of the input fuzzy sets and sampling are investigated results from the practical implementation are presented. The main contributions in this chapter are a study of how the input resolution of the chosen fuzzy input sets affect the controller performance and a comparison of the processing performance of the fuzzy controller algorithm when implemented sequentially and in parallel. The robustness and stability of the fuzzy controller are also verified.

Chapter 7 summarises the work presented in this thesis, and draws conclusions about practicalities of the proposed solution to real-time robots.

Chapter 8 gives recommendations for further work.

Appendix A is the ACSL source code for the simulation of the prototype robot and fuzzy controller.

Appendix B gives the C++ source code for the computation of fuzzy sets used in the simulations and practical implementation of the controller.

Appendix C gives the configuration file and C source code for the implementation of the fuzzy controller in real-time on a network of transputers.

Appendix D gives the datasheets for the transputer based data acquisition cards.

Appendix E provides the datasheet for the Harmonic Drive actuators used on the prototype.

Appendix E shows the design and screen dump of the Labview data acquisition program used to investigate high sample rates for the practical implementation of the controller.

References

[1] Klafter R. D., Chielewski T. A., and Nagin M., editors. *Robotic engineering : An integrated approach*, chapter 1.3: Classification of Robots. Prentice Hall, Inc., 1989.

[2] Gateff P. A. and Abbott J. C. Laboratory robotics: Applications in the materials science laboartory. In Strimaitis J. R. and Hawk J.L., editors, *Advances in Laboratory Automation - Robotics*. Zymark Corporation, Hopkinton, MA., 1985.

[3] Whitney D. E., Book W. J., and Lynch P. M. Design and control considerations for industrial and space manipulators. *JACC*, 1974.

[4] Andeen G. B. *Robot Design Handbook*. SRI International, 1988.

[5] Wilson M. Industrial Robotics in the UK. In *Proc. of the International Workshop on Advanced Robotics and Intelligent Machines*, Salford, Manchester, 1996.

[6] Editor. Robots move in to tackle heavier weights on building sites. *New Scientist*, 123(1680), 1984.

[7] Wanner M. C. Design of a manipulator with very large reach for applications in civil engineering. In *Robomatrix Reporter Annual 1988*, 1988.

[8] Surdhar J. S. and Korhonen J. A fuzzy PD controller applied to a long reach manipulator. In *Middlesex University Faculty of Technology Technical Report Series ,MUCORT - Middlesex University Research Conference*, Middlesex University, London, 1995. ISSN 1362-2285.

[9] Book W. J. Structural flexibility of motion systems in the space environment. *IEEE Transactions on robotics and automation*, 9(5):524–530, 1993.

[10] Davies J. B. C. Elephants trunks an unforgettable alternative to rigid mechanics. *Industrial Robot*, 18:29–30, 1991.

[11] Brady M., Hollerbach J. M., Johnson T. L., Lozano-Perez T., and Mason M. T., editors. *Robot motion: Planning and control*, chapter 1: Introduction. MIT Press, 1982.

[12] Zalucky A. and Hardt D. E. Active control of robot structure deflections. *Journal of Dynamic Systems Measurement and Control - Transactions of the ASME*, 106:63–69, 1984.

[13] Driessen F. P. G., Lucassen F. H. R., and Van de Ven H. H. *A three dimensional measuring system*, pages 309–313. Number 3. 1986.

[14] Azdhari A., Chaloub N. G., and Gordaninejad F. Dynamic modelling of a revolute-prismatic flexible robot arm fabricated from advanced composite materials. *Journal of Nonlinear Dynamics*, 2:171–186, 1991.

[15] Thompson B. S and Sung C. K. A variational formulation for the non-linear finite element analysis of flexible linkages: Theory, implementation and experimental results. *ASME J. Mech. Trans. Automat. Design*, 106(4):482–488, 1984.

[16] Ramirez R. B. Design of high speed graphite composite robot arms. *ASME paper No. 84-DET-50*, 1984.

[17] Liao D.X., Sung C. K., and Thompson B. S. The design of flexible robotic manipulators with optimal arm geometries fabricated from composite laminates with optimal material properties. *The International Journal of Robotics Research*, 6(3):116–130, 1987.

[18] Tzes A., Englehart M., and Yurkovich. Input preshaping with frequency domain information for flexible link manipulator control. In *Proc. AIAA Guidance, Navigation and Control Conference*, Boston, MA, 1989.

[19] Magee D. P. and Book W. J. Control and control theory for flexible robots. *Journal of the Society of Instrument Control Engineers*, 4:309–317, 1993.

[20] Desoyer K., Kopaceck P., Lugner P., and Troch I. Flexible robots - a survey. In *Proceedings of the IFACS/IFIP/IMACS Symposium*, pages 23–34, 1989.

[21] Roth Z. S., Mooring B. W., and Ravani B. An overview of robot calibration. *IEEE Journal of Robotics and Automation*, RA-3(5):377–384, 1987.

[22] Manganas A. Tip stabilisation of base compliant manipulators, 1993.

[23] Kyle S. A. Optical methods for calibrating and inspecting robots. *Computing and Control Engineering Journal*, 6(4):166–173, 1995.

[24] Sweet L. M. and Good M. C. Redefinition of the robot motion control problem. In *IEEE control systems magazine*, pages 18–25, 1985.

[25] Sharon A. Hogan N. and Hardt D. The macro/micro manipulator: an improved architecture for robot control. *Robotics and Computer-Integrated Manufacturing*, 10(3):209–222, 1993.

[26] Gevarter W. B. Basic relations for control of flexible vehicles. *AIAA J.*, 8(4):666–672, 1970.

[27] Cannon R. H. and Schmitz E. Initial experiments on the end-point control of a flexible one-link robot. *The International Journal of Robotics Research*, 3(3):62–75, 1984.

[28] Lewis J. The design, construction and testing of a laser guided robot arm, MSc dissertation, Middlesex University, London, England. 1991.

[29] Scott A. G. Position control for machines. In *UK Patent Application No. 8908755*, 1989.

[30] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[31] Harashima F. and Ueshiba T. Adaptive control of a flexible arm using the end point position sensing. In *Proceedings of the Japan - U.S.A. symposium on Flexible Automation*, pages 225–229, 1986.

[32] Bailey T. and Hubbard J. E. Distributed piezo-electric active vibration control of a cantilever beam. *J. Guidance and Control*, 8(5):605–611, 1985.

[33] Chernousko F. L.and Rogov N. N. Optimal control of a robot with electric drive and elastic element. In *Robot Control 88, IFAC Symposium*, pages 91.1–91.5, 1985.

[34] Mayr 0. The origins of feedback control. *Scientific American*, 223:110–118, 1970.

[35] Tzafestas S. AI techniques in control:An overview. In *Int. Symposium on AI, Expert Systems and Languages in Modelling and Simulation*, 1987.

[36] Albertos P. Fuzzy controllers. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of Artificial Intelligence in Process Control*, pages 343–367. Pergamon Press, 1992.

[37] Kosko B. *Neural Networks and Fuzzy Systems*, chapter 1, page 19. Prentice Hall, UK, 1992.

[38] Kosko B. *Neural Networks and Fuzzy Systems*, chapter Preface, page xxvi. Prentice Hall, UK, 1992.

Chapter 2

End-point control for flexible link manipulators

2.1 Non-minimum phase transfer functions and spillover

In the control of flexible manipulator arms with non-collocated actuators and sensors, engineers are faced with the problem of working with systems that have a non-minimum phase transfer function. A system is said to have a non-minimum transfer function when its zeros lie in the right hand (RH) s-plane [1].

RH zeros can introduce a large phase lag between the control signal and measured output signal. When the phase shift is in excess of 180°, conventional passive compensation techniques cannot be applied. Sophisticated controllers, such as optimal (Linear Quadratic Regulator) LQR or (Linear Quadratic Gaussian) LQG based on pole-placement, can be used to compensate for the phase shift provided the model of the system is highly accurate. However, such models are not easily achieved in practice. Large phase lags are common in flexible arm systems that are often modeled as having an infinite number of vibration modes. Each successive higher mode can contribute to further 90° phase lags. In addition to the large phase lags in the system there is a problem due to the unpredictable behavior of the un-modelled higher modes especially when a truncated model is used. This phenomenon is known as 'spillover' and may cause controller instability [2][3].

The physical interpretation of a non-minimum phase system in terms of a robot with a flexible link is that there is a finite delay, due to propagation of vibrations, between sensing at the tip of an arm and the applied control action. Spillover is attributed to unpredictable interactions between 'controlled' vibrations or modes that have been measured and 'uncontrolled' unmeasured vibrations that may be propagating along the arm. The latter case, referred to as observation spillover as opposed to control spillover, is more critical for control purposes [4].

Many workers have concentrated their efforts to study and find a solution to the control of manipulators with flexible links with non-collocated sensors [5][6][7]. It has been found that for flexural deformation of an elastic beam the location of the zeros is dependent on the displacement of the sensor from the actuator [5]. Also, the frequency at which a system transfer function becomes non-minimum phase decreases with sensor/actuator separation [8].

Various methods have been investigated in order to solve the problem of finding a suitable minimum phase transfer function. These range from accurate pole placement, to cancel the effects of the right-hand-side zeros using optimal control techniques [9], to derivation of models with minimum phase transfer functions, by using modified outputs or alternative non-collocated measurements [10][11].

The following chapter reviews how different workers have solved the problem of controlling systems which are potentially non-minimum phase, particularly, manipulators with flexible links that are based on non-collocated sensors and actuators. Emphasis will be placed on experimental work utilising sensors for deflection measurement. Cases with acceleration or force sensors will only be examined in the context of non-collocation. The aim of the study is to define a foundation for the

design of a new controller based on the experience of other workers. The review also serves as a benchmark against which performance of the new controller will be compared.

Controllers based on non-collocated sensing can be divided into two categories: -

1. Those in which the sensor is positioned locally along the arm between the actuator and tip - i.e., *local deflection measurement*,
2. Those that employ sensing at the tip of the arm - i.e., *end-point deflection measurement.*

Both categories are reviewed as they deal with a similar problem of controlling a system which has a potentially non-minimum phase transfer function. Tip Feedback Sensor (TFS) based control described in this thesis is classified under the latter category. The emphasis of the review is on experimental work as many theoretical results are made more complex or even invalidated by the use of inaccurate models and the effects of non-minimum phase caused by non collocation of the sensor and actuator [12]. Implementation of the controller in real time introduces further constraints on the controller design and could even invalidate the approach.

2.2 Local deflection measurement

Kanoh *et al.* [13] reviewed modeling and control problems of flexible robot arm systems and acknowledged that controllers for manipulators with flexible links, based on reduced order models, suffered from spillover. They suggested that spillover could be contained by either filtering off the higher un-modelled modes in the controller output or by the use of more accurate models and sophisticated controllers. Non-linear drive flexibility [14][15], in the case of a harmonic gear reducer, coulomb friction and the current limits of the amplifiers were outlined as sources for modelling error and their inclusion in the model was recommended in order obtain more accuracy.

In an earlier paper Kanoh and Lee [16] discussed the effect of sensor location on controller stability and established that by placing the sensor in the region where the 2^{nd} order derivatives of the modes is positive the effects of spillover can be contained and closed loop stability ensured. This criterion required that the sensor be placed very close to the actuator.

Fukuda and Kuribayashi [17] proposed a state feedback control method and a model based on a Newton-Euler formulation with modal expansion. The controller used the first three modes of vibration which were determined by using strain gauges mounted on the arm. However, there were problems with accuracy and overshoot due to the strain gauges being able to detect only local distortion of the arm [18].

Wang *et al.* [19] used a laser and mirror arrangement to measure the slope of a 50 x 2 x 1000 mm arm. Stability of the controller was maintained by ensuring the sensor was located in the region along the beam where the sign of the deflection and hub angle measurements were positive, thus ensuring that the system was minimum phase. The controller was able to cope with different payloads but this was achieved at the expense of increased settling time in the step response. The effect of increasing the payload to 90% of the arm mass caused the settling time to increase by over 300%. The use of the slender 'theoretical structure', which had a much smaller thickness compared to width, allowed use of a simpler model where the effects of warp and shear could be ignored in controller design [1].

Wang's method is susceptible to the same problems as the configuration of Fukuda and Kuribayashi above, since only local measurements of the slope are being made. In addition, the external sensing arrangement places constraints on the practicality of the system as obstacles may interrupt the measurement signal, especially in a cluttered environment. Also, gravitation effects were ignored in these experiments as motion of the arm was restricted to the horizontal plane.

Luo [21] verified experimentally the stability of a controller which was based on measurements from strain gauges mounted near the arm hub. Limitations of this method were an inability to cope with payload variations and, as with most controllers based on local deflection measurement,

[1] Warp and shear introduce further torsion vibration modes [20] which may be excited by the control and cause a degradation in control performance [4].

requirement of a highly accurate model to measure the required modes correctly for sensor location. The first limitation was overcome by using adaptive control [22]. The design of the experimental apparatus permitted the effects of gravity and torsion to be ignored.

Swevers *et al.* [23] used DIOMEDES (Laser DIode Optical System for MEasuring Structural DEflectionS), as described in [24], and a combination of feedforward and feedback control with an accurate model of a flexible one-link robot. A low pass Butterworth filter with a cut-off of 10 Hz was used at the control output to prevent spillover as the model used accounted only for the first mode of vibration which was measured at 8.4 Hz. This scheme resulted in accurate tracking of a tenth order polynomial reference trajectory. Emphasis was laid on the use of an accurate model based on a frequency domain identification approach [25]. Gravity and torsion effects were ignored in these experiments. The work of Swevers *et al.* shows that the effects of spillover can be avoided by filtering of the higher modes of vibration. However, there are limitations in this work in that it relies heavily on an accurate model of the system. Accurate modelling was facilitated through the use of a 'theoretical' structure which did not suffer vibrations due to torsion. This method cannot be guaranteed to give accurate positioning as the deflection measurements are not tip based. Different tip payloads would further complicate the design of the model based controller.

2.3 End-point deflection measurement

The previous section reviewed non-collocated methods of control where the sensor was not placed at the tip of the manipulator arm. Methods based on 'local deflection' measurement are susceptible to inaccurate end-effector positioning. In contrast end-point sensing can be can be more accurate despite link deformations and modelling errors. The underlying philosophy of non-collocated end-point control is epitomised by Demeester and Van Brussel [24] who declared:

> *'To control the structural deflections and vibrations of flexible manipulators, knowledge of the position of the end-effector or of the end-point of the flexible link is essential. Knowing the position of the end-effector, all deflections, backlash and other error sources can be compensated'.*

This profound statement is justified by the pioneering work of Cannon and Schmitz who published experimental results [9] based on the control of very flexible manipulators where sensors are not collocated with the actuator. In doing so they went against the norm laid by Gevarter [26] who showed that non-collocated controllers are unstable at high gains which limits achievable system bandwidth and is detrimental to response time and dynamic accuracy of the system.

Cannon and Schmitz's generic experimental work demonstrated the feasibility of direct tip position control and showed the potential effectiveness of using optical end-point sensing systems. Good stability and a response three times faster than the natural cantilever period of the system was achieved using a reduced-order model and linear quadratic gaussian (LQG) controller with an optical tip sensor. They stressed that a good model of the system dynamics and sophisticated control algorithms were essential to prevent the effects of spillover which may arise due to use of a reduced order or truncated model of a flexible arm. Their work relied on a very accurate model of the system being controlled. The modelling was aided by the fact that the flexible manipulator arm was designed to be flexible only in a single plane. In real practical systems link flexibility is not necessarily restricted in this way.

Control systems which utilise optical end-point sensing can be classified broadly under the following different groups.

Transfer function modification through hardware These systems attempt to change the non-minimum phase transfer function of the flexible structure by reconfiguration of the physical system. This may be achieved by judicious sensor placement as described in [16] and [19], or by use of some form of tip actuation.

Transfer function modification through software These systems attempt to change the non-minimum phase transfer function of the flexible structure by reconfiguration of model through

use of alternate output measurements [27]. The aim of achieving a minimum phase transfer function is that passive control instead of more complex and sophisticated methods can be used. This is attractive from the control point of view especially in real-time situations where computation time of the control algorithm is critical. Also stability analysis can be done more easily for passive controllers using standard tools and methods.

Controllers requiring accurate models These methods require highly accurate model of the system in order to implement the control law successfully. Examples of control techniques requiring accurate models are inverse dynamic control, use of modified inputs to the system in order to reduce tip vibrations and adaptive control which require identification or estimation techniques in order to derive a controller that can cope with the parameters of the system.

Tip regulation and gravity compensation Most experimental systems restrict motion of the manipulator to the horizontal plane hence ignoring the effects of acceleration due to gravity. These experiments have been motivated by a space application based impetus. The effects of gravity can cause a significant difference to system performance and must be considered when designing a control system for a flexible manipulator which is not used in applications where gravity is negligible.

Artificial intelligence methods These differ from the more conventional methods categorised above in that the controller incorporates fuzzy logic, neural networks or expert systems. Usually these methods are employed in the modelling and control of non-linear processes that can not be described easily with a mathematical model.

2.3.1 Transfer function modification through hardware

Zalucky and Hardt [28] use a 'straightness servo' to achieve accurate deflection compensation despite payload variation. Their system comprised two concentric beams, one acting as the manipulator link and the one to carry the bending loads, an optical displacement sensor and a straightness actuator mounted on the rigid beam. Although high accuracy has been achieved, in spite of payload variation, the physical structure of the system places constraints on the practicality of the scheme as the inner stiff beam and compensating actuator impose a further payload and cost. Also, a second sensor-actuator pair will be required to compensate deflections in the horizontal plane. The effect of bending errors and vibration caused by torsion was not addressed in the experiments.

Davis and Hirschorn [29] Considered the problem of spillover due to design of a controller with a mode truncated model of the system. Their approach was to formulate a model which included a distributed model of the link dynamics combined with a discrete model of the actuator and load dynamics. The model was used to show that use of a conventional inverse system tracking controller had non-minimum phase characteristics. Their solution was to reconfigure the hardware in order to collocate the tip sensor and actuator. No experimental results were presented but it is clear that the coaxial system will suffer from the same impracticalities as in the system of Zalucky and Hardt described earlier.

Sharon and Hardt [30] use a micro-manipulator mounted on the end of a large manipulator to compensate for the effects of unmeasured structural deflections and poor actuator servo resolution. The large manipulator is used for the faster gross motion and the micro-manipulator for the fine motion once the large manipulator has arrived near the desired location. Operation of the micro-manipulator relied on tip position feedback measured using a form of external optical sensing. A simple proportional controller was used to control the micro-manipulator. Stability of the controller depended on the structural frequency of the beam. A stable, well damped, system was achieved for a system bandwidth lower or equal to the effective structural frequency of the larger robot. This means that only correction of static errors is possible using proportional control. For higher system frequencies, the behaviour of the proportional controller was unpredictable. A parameter adaptive controller and observer were recommended to overcome this problem. Fig 2.1 illustrates the experimental set up used by Sharon and Hardt.

Almost a decade later Sharon *et al.* [32] presented results of a study of the effect of the micro-manipulator on the stability and performance of the macro/micro manipulator. They showed that

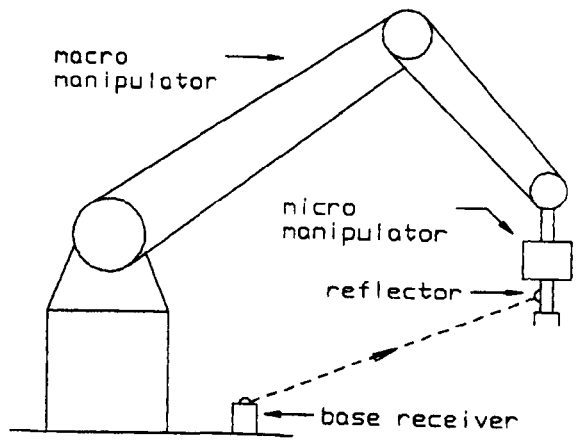

Figure 2.1: Micro-manipulator system [31].

stability of the system was dependent on the dynamic coupling between the micro-manipulator and macro-manipulator. They proposed a controller designed in the physical domain [33], where the macro/micro-manipulator is regarded as a mechanical transmission line, reducing the problem of control to one of impedance matching [34]. This controller was shown to be stable regardless of the effects of dynamic coupling in the system. Although the stability problem was solved as described above the micro-manipulator of Sharon *et al.* remains impracticable especially in cluttered environments. The tip position for the micro-manipulator is sensed by external means and in some cases these measurements may not be possible due to obstacles, which may shield the base receiver from the sensor, or due to the environment itself, for instance in painting applications where the sensor may be covered by spray. A further disadvantage of the proposed system is that two separate controllers are required for the macro and micro-manipulator. The effects of gravity were ignored as the experiments were carried out restricting motion only to the horizontal plane.

Chalhoub and Zhang [35] solved the measurement problem of Sharon and Hardt by incorporation of a dual axis sensor comprising of a laser/photo detector arrangement to detect deflections in the arm . The principle of deflection measurement is similar to that used by Lewis [31][36]. However, Chaloub and Zhang use the measurement for positioning a dual axis micro-manipulator in contrast to positioning the arm tip as done by Lewis. The micro-manipulator controller (MMC) was designed independently from the controller for the flexible link. An improvement in control performance was shown for a combinations of the MMC and a rigid body controller (RBC), and a MMC and rigid body and flexible motion controller (RFMC), in contrast to control without the MMC. These experiments proved that it was possible to reduce the positional inaccuracy of a gripper induced by vibrations of the compliant beam. However, the errors corrected were of a static nature i.e., vibrations due to the system dynamics were not damped and the gripper appeared to be chattering about the end position. The experimental set up allowed the effects of gravity to be ignored in these experiments. A further limitation is that errors due to detector tilt, through bending of the loaded host arm, could not be detected. Different tip payloads were not considered in these experiments.

Chiang *et al.* [37] used a micro-manipulator to increase the end-effector position control bandwidth. Fig. 2.2 depicts the experimental arrangement used by Chiang *et al.* Optical sensing was only used to indicate the desired end-point of motion. If the host arm were to move too quickly, the bulb would swing out of range of the position demand photodiode and micro-manipulator control would be lost. These experiments suffer the same external sensing limitations as those of Sharon *et al.*

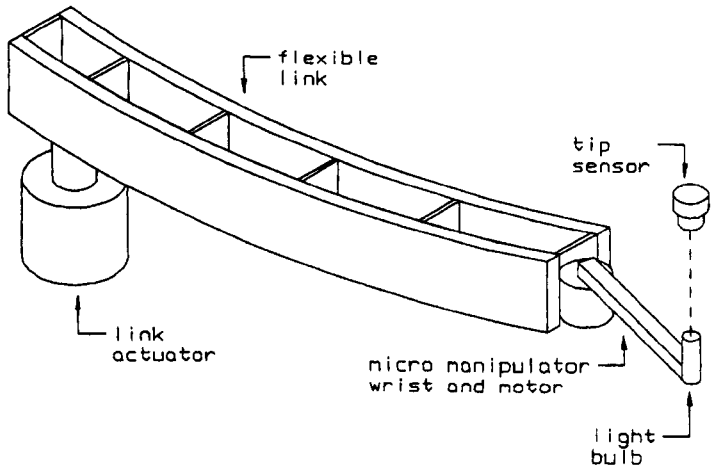

Figure 2.2: Micro-manipulator system of Chiang *et al.* [31].

2.3.2 Transfer modification through software

Wang and Vidyasagar [19] investigated the use of a modified output measurement in the end-point control of a flexible link manipulator. Their studies showed that by using the *reflected* tip position measurement as opposed to the actual measurement, it was possible to achieve a model of the system with a minimum phase transfer function. A reflected output measurement was obtained by reversing the polarity of the measured end-point position of the flexible link. Pota and Vidyasagar [10] extended this work to include proof of the model satisfying the passivity theorem which states that any strictly passive cascade compensator with a finite gain will result in a L_2 stable system [38]. In particular their results alluded to a passive PD controller. Simulations showed that a flexible link could be brought to rest in response to a 1 radian step in under 1.5 seconds.

Wang and Vidyasagar [39] obtained experimental results that were in agreement with the simulations of Pota and Vidyasagar. A later study [40] showed that passive control could be used even as the manipulator is made more rigid where the significant modes of vibration can be high enough to exclude the use of digital controllers because of sampling time limitations. This work is significant in that it shows that use of a *reflected* output measurement and PD control gives stable non-collocated control. The work of the Waterloo University group above did not address the problem of different payloads applied at the arm's tip.

2.3.3 Controllers requiring accurate models

Bhat *et al.* [41] used a Laplace transform technique for point-to-point[2] position control of a flexible beam. This controller was based on the physical requirement that any strain energy trapped in the elastic modes of the structure must be totally relieved at the end of the desired rigid body displacement so as to eliminate residual vibration [42][43]. A tenth order truncated version (rigid body and first four flexible modes) of the physical model was used for evaluating the open loop control. Smallest residual vibration at the tip and quickest step response is achieved with a polynomial input as opposed to bang-bang and minimal norm control inputs. In a later publication [44], they extended this work to feedforward control. In this configuration the problem of eliminating residual vibration reduced to one of designing an open loop control input for a closed loop system. Experiments confirmed polynomial feedforward inputs fared better than bang-bang inputs. The experimental apparatus consisted of a slender 740 x 50 x 0.8 mm spring steel beam and tip displacement is measured using a laser interferometer. Motion of the arm was restricted to the horizontal plane and it is evident that torsion effects were ignored as the thickness of the beam structure was much smaller than the width.

[2] all the states $\vec{x}(t)$ are specified at $t = 0$ and $t = T$ for a dynamic system governed by a linear ordinary differential equation with time-invariant coefficients

Grieco *et al.* [45] defined the trajectory for the manipulator based on the assumption that the link is rigid. The control was based on two loops, a tight inner loop for positioning of the 'rigid' arm and a second outer loop, which relied on a camera measured tip position to close tracking errors between the tip and hub. The problems associated with non-collocation are avoided as the implementation of the controller modifies the motor hub set point instead of the error signal coming directly from the tip. This in principle, is similar to the method of input preshaping. The experimental apparatus consisted of a slender beam of length 0.66 m and mass 0.059 kg attached to an actuator. Movement was restricted to the horizontal plane. These experiments did not consider the effects of payload variations on controller performance.

Harashima and Ueshiba [18] and Harashima *et al.* [46] used an adaptive controller based on an autoregressive (AR) model with dead time. Tip deflections measured with a non-collocated end-point deflection sensor, a charge coupled device (CCD) camera, were used for on-line model parameter adjustment. Experiments confirmed that fast positioning without overshoot could always be achieved in spite of any parameter changes such as payload variation. The method of control was confined to a single axis and the effects of gravity were unaccounted for as motion was in the horizontal plane.

Rovner and Cannon [47] applied an adaptive algorithm based on Astrom's [48] self-tuning regulator to an arm similar to the earlier experimental arm used by Cannon and Schmitz [9]. This method relied on a faster time-domain identification of the system model which allowed the control design procedure to be automated. Accurate pick and place position control of the manipulator with different tip loads of 0 g, 125 g and 250 g was achieved for motion in the horizontal plane. In these experiments the tip loads applied are relatively small in comparison to arm weight. The increase in moment of inertia of the manipulator from the unloaded case to the 250 g case is approximately 50%.

Alder and Rock [49] developed a control technique based on on-line identification of the system using a subspace fitting technique which allows identification without assumption of system order and a self tuning regulator. Other workers [50][51][52][53] have also designed payload adaptive controllers based on the self-tuning regulator solution. The difference between the work of Alder and Rock and the previous workers is that unknown payload dynamics are considered in contrast to the assumption that the payload is a rigid body. The controller provided precise end-point control of a flexible manipulator while simultaneously damping internal oscillations of a pendulum suspended at the tip of the arm. The experimental set-up is depicted in Fig. 2.3. As the experiments were designed with a space application in mind the effects of gravity were also ignored.

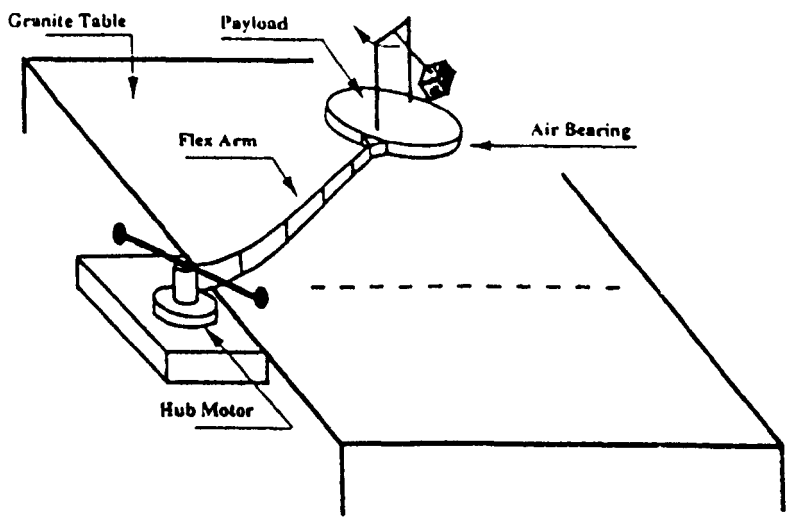

Figure 2.3: Experimental set-up used by Alder and Rock [49].

2.3.4 Tip Regulation and gravity compensation

Mulders *et al.* [54] showed that a flexible beam may be actively stiffened by the use of end-point deflection sensing and piezo-electric actuators at the root of a fixed beam. This method was likened to the manner in which an angler lifts his rod to compensated for its deflection. Fig. 2.4 illustrates the schematic diagram of the stiffening arrangement. Stiffening was achieved without the use of a dynamic model of the system and the measured deviation was the sole input to the PID controller used.

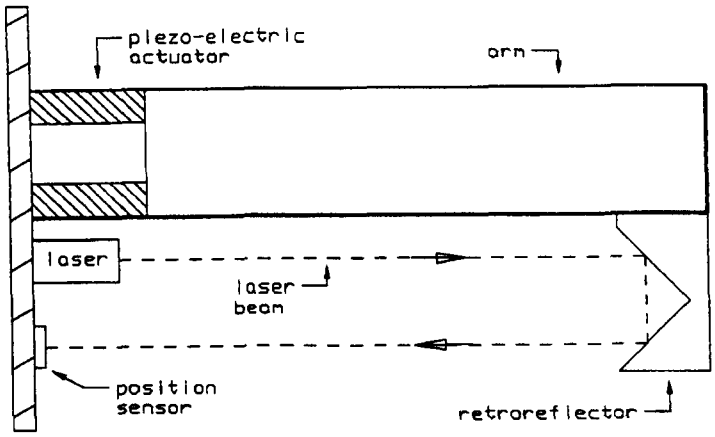

Figure 2.4: Schematic of stiffening arrangement [31].

De Luca and Panzieri [55] show that compensation for gravity for accurate placement of robots with flexible links can be obtained via a simple iterative control procedure that does not require knowledge of the arm mass and stiffness (i.e., no dynamic model of the arm is required). They extend this PD based control method to direct control of the tip position. Stability of the system is ensured by placing a lower bound condition on the magnitude of the proportional gain in the PD control part. This method is used for regulation of the tip at a specific set point and it is not clear whether vibrations are actively compensated. De Luca and Panzieri point out that asymptotic stability of PID controllers has not been proven for flexible arms and that addition of a an integral term to compensate for gravity terms may cause saturation and windup problems due to the non-linear nature of the robot dynamics. Lewis [31] also observed similar effects, leading him to the choice of using PD control as opposed to PID control.

2.3.5 Artificial Intelligence based methods

Garciabenítez *et al.* [56] applied a rule-based supervisor, in a two level hierarchical control strategy, to improve the performance of existing techniques for tip position control of a two link manipulator with flexible links. The upper level consisted of a feedforward rule-based supervisor that incorporated fuzzy logic for controller and parameter selection. The lower level in the hierarchy consisted of three different controllers based on input shaping. The first controller in the lower level comprised of input shaping alone [57] whilst the other two were made up of a combination of input shaping and shaft position end-point acceleration feedback [58]. The role of the supervisor was divided into three subsystems with different tasks:

1. Selection of the controller that best minimises end-point vibrations in the links depending on manipulator configuration.

2. Determination of the amplitude and time duration for parameters of the lower level controllers in order to minimise torsional vibrations in the links.

3. Adjustment of the input parameters that shape the actuator reference voltages in order to minimise link bending vibrations.

The first subsystem was a simple rule based system which selects a controller based on the joint angles of the manipulator. The second subsystem was a 36 rule multi-antecedent (two input variables), multi-consequent (two output variables) fuzzy system, which can be decomposed to a 72 rule multi-antecedent single-consequent (2 input variables and 1 output variable) system. The third is a 9 rule multi-antecedent multi-consequent fuzzy subsystem which can be decomposed to an 18 rule system.

Although faster settling times and smaller overshoots are achieved using the supervisor and existing conventional controllers there are several drawbacks to the hierarchical method, these are:-

1. Additional computational overheads are incurred due to the rule base and inference mechanism for the fuzzy subsystems. For real-time applications the advantage of fewer fuzzy rules far outweighs the importance of heuristic decision rules [59].

2. The use of end-point acceleration feedback cannot guarantee accurate end-effector positioning since errors due to permanent link deformations cannot be corrected.

3. Gravitation and torsion effects have been ignored in these experiments - these would further complicate controller design.

4. A precise dynamic model of the system is required for the lower level input shaping controllers as these operate in open-loop.

Moudgal *et al.* [60] devised a fuzzy model reference learning controller (FMRLC) [61] for a flexible robot. This work was an expanded version of that reported by Garciabenítez *et al.* [56]. The same experimental test bed, consisting of a counterbalanced two link flexible manipulator restricted to move in the horizontal plane, was used. Dimensions for the links were 750 x 127 x 23 mm for the shoulder and 500 x 38 x 1 mm for the elbow. Optical encoders were used to sense joint position and accelerometers were used to sense tip accelerations. The idea of supervising/tuning lower level controllers in a hierarchy is extended by using a higher level learning mechanism to synthesise and tune a lower level rule-based controller. The advantage of using a lower level fuzzy controller, as opposed to input shaping controllers, was that an accurate dynamic model of the system was not required. Moudgal *et al.* showed that it was possible to achieve no overshoot at the end-point in response to a slew input with the lower level fuzzy controller.

They later showed that a fuzzy learning controller, inspired by Procyk and Mamdani's [62] linguistic self organising controller, was able to match the control performance of the manually synthesised rule-base. Rules were automatically derived through a learning mechanism which adjusted the knowledge base of the fuzzy controller. The learning mechanism relied on a reference model, specifying the transient behaviour of the closed loop response, and a fuzzy inverse model of the system which changed plant command inputs according to deviations between the actual plant behaviour and that of the reference model. The learning ability of the controller also made it suitable for tuning its response to an increased payload (30 g) and a slightly more damped response was obtained than for the controller with manually synthesised rules. The idea of a fuzzy controller which adapts its behaviour to a variation in payload is attractive because it takes away the tedium of having to manually derive the rule base for the controller and control is performed according to a suitable reference model. However, in the case of Moudgal *et al.* the inverse model of the system is also rule based and its design was carried out entirely on a heuristic basis from detailed observations of the plant. This detracts from making rule design for the fuzzy controller simpler and it also has the effect of increasing the number of rules to be processed which has the consequence of an increase in the controller sampling time (use of the fuzzy inverse model introduced a further 686 rules in addition to the 464 for the fuzzy controllers for both joints). A sampling rate of 15 ms was achieved using an Intel 386SX processor running at 25 MHz. This scheme also suffers from the first three limitations as the experiments of Garciabenítez *et al.* described earlier.

Wu *et al.* [63] compared the performance of a self tuning regulator and fuzzy logic controller for tip regulation of a single flexible link. The control strategy comprised of a PID controller in a tight inner loop about the hub actuator and either the fuzzy controller or self tuning regulator in a second loop utilising tip acceleration measurements to regulate tip vibrations. The experimental set up consisted of a 300 x 15 x 1 mm slender beam attached to a DC motor. The fuzzy logic

vibration controller, with 49 heuristically designed rules, was shown to have shorter settling times for the tip acceleration than the self tuning regulator. This result prevailed even in the presence of a 100% increase in the mass of the link (from 13 g to 26 g). The self tuning regulator was shown to have slightly smaller steady state error at the hub than the fuzzy controller. It was pointed out that the fuzzy controller did not rely on a precise mathematical model of the system was simpler to implement than the self tuning regulator. Effects of gravity and link torsion were not considered in these experiments.

Takahashi and Yamada [64] reported the first experimental work involving neural network control of a single link flexible arm. The experimental set up consisted of a slender lightweight arm (length 400 mm and mass 19 g) attached to the shaft of a DC servo motor. Motion was restricted to the horizontal plane. End-point angular position was estimated from strain gauge measurements near the hub of the arm. The controller was shown to effectively control the tip angular position of the arm with small overshoot despite an increased payload of more than two and a half times the arm mass. Settling times from the unloaded to loaded case increased from 1.8 s to 3.2 s for a tip angular displacement of 1.5 radians. These experiments show the viability of devising a learning based controller in order to compensate for the nonlinearity and/or uncertainties of the system. However, the method of sensing cannot guarantee accurate end-point positioning and should strictly be classified as a local-deflection method. It has been included in this section because of its importance as one of the first practical implementations of neural networks for the non-collocated control of a flexible link. The controller design did not take into consideration the effects of gravity and link torsion.

Kubica and Wang [65] developed a fuzzy logic controller (FLC) to control the rigid body and first flexural mode in a single link flexible arm. The controller comprised of a fuzzy hub-based controller with additional rules for vibration suppression. The hub angle and its derivative formed inputs to the rigid body controller and the tip deflection and its derivative formed inputs to the vibration suppression part. The control input to the actuator was comprised of the sum of the outputs of the rigid body and vibration controllers. A total of 108 rules were required for the complete controller. In comparing their results to the passive PD control method by Wang and Vidyasagar [40] (under similar experimental conditions) the FLC was shown to have a significantly faster settling time (1.1 s as opposed to 1.5 s) and elicited less high frequency vibration. They were unable to confirm the stability of this system.

Lee *et al.* [66] designed a fuzzy controller to control the position of a manipulator with a single flexible link. This model did not require the formulation of a mathematical model of the system. The experimental arrangement consisted of a single flexible link rotating in the horizontal plane. The arm was designed so that deflections were only allowed in the plane of movement. Different payloads could be attached to the arm. An external ultra-sonic 3D position sensor was used to measure the tip deflections and hub angular position was measured using the motor encoders. The motor could be controlled either manually by a human operator using a joystick or via a computer algorithm. The manual control was used for the purpose of deriving fuzzy rules for the fuzzy controller. The idea behind this approach was that a human operator has the ability to move the arm from an initial point to a specified point while suppressing its vibrations. The knowledge gained from these experiments was then transformed into fuzzy control rules. The antecedant or input part of the fuzzy controller consisted of measurements of the tip position and its derivative and the consequent or output control variable was the speed of the motor. A total of 49 fuzzy rules were used to implement the controller. These rules were defined via observations of the manual controller and through trial and error simulations. Gaussian shaped sets were used for both the antecedant and consequent. The fuzzy inference engine was built using the *Minimum Operation* rule and defuzzification was carried out using the *Center of Area* method (this type of defuzzification generally yields a smoother output response [67]). Simulation results showed that a 1000 x 50.8 x 2.54 mm arm with a payload of seven times the arm mass could be brought to rest in under 2 seconds in response to a 80 cm ramp input. Lee *et al.* experimented with combinations of antecedants and found that a combination of the position and its rate provided the best control results. This type of controller can also be referred to as a direct fuzzy PD controller [68]. They noted from their simulations that the fuzzy scale factors and sample rates had a significant effect on system response. These parameters would greatly influence the design of a practical controller.

At the time of the publication of their paper, experimental results were yet to be achieved.

2.4 Discussion

The non-minimum phase characteristic of end-point control has spurred many different solutions to the control of manipulators with flexible links. The solutions range from different hardware configurations, filtering off higher vibration modes, through to derivations of different models of the system in software. The majority of these solutions reviewed seek to achieve a minimum phase transfer function that will lead onto the design of a stable controller. Exceptions are the adaptive control methods relying on parameter estimation or online system identification and some AI based control methods.

Mulders *et al.* [54] and De Luca and Panzieri [55] have shown that it is possible to regulate the end-point of a flexible link, despite the effects of gravity, using simple end-point feedback and classical control. Achieving a minimum-phase transfer function was not an issue in these experiments since a model was not used by the controller. Their approach is very similar to that of TFS based control except that trajectory definition for the end-point of their manipulators was done independently from the regulation task. It is not desirable to have two competing separate control loops, i.e., one for position control and one to compensate for link deformation, as coupling between them can lead to instability [31].

The adaptive control methods as investigated by the pioneers of end-point control, Cannon and Schmitz and their fellow researchers at the Stanford University's Aerospace Robotics Laboratory (ARL) including Rock, Alder and Rovner all advocate the use of a highly accurate model. Impressive results have been achieved by these workers in terms of end-point control of flexible link manipulators with different and even dynamic payloads. In all these experiments the accuracy of their model was augmented by end-point measurements. However, the experimental work of these researchers has been reliant on a mathematically well behaved and custom built experimental rig in order to facilitate modelling.

The model based adaptive approach is not available in the case of the Middlesex prototype robot. There are two reasons for this:-

- The robot is fraught with non-linearity and the round cross section of the link can be the source of infinite torsion vibration modes that will be difficult to accurately model.
- The use of model based control would detract from one of the aims of this research which is to retain the simplicity of the TFS based approach.

The approach adopted in this thesis aims to bring the advantages of adaptive and non-linear control to the Middlesex prototype via the remaining option i.e., via fuzzy control which is an AI based method.

AI control techniques are a more recent addition to the types of controllers being applied to control flexible manipulators with flexible links using end-point sensing. These methods do not require the formulation and validation of a mathematical based model of the system and have been shown to be useful for control of plants that exhibit non-linear and uncertain behaviour. The work of Lee [66] and Wu [63] has shown that fuzzy control can be used for end-point control of flexible link manipulators that have payloads attached to the arm. These findings serve to underline the choice of fuzzy control in this thesis. Both Lee and Wu used 49 fuzzy rules in their controllers and their experiments were carried out on 'theoretical' structures that only allowed vibrations in the horizontal plane. It is possible that use of a round arm cross section, and extension of control to a second axis, such as the one on the Middlesex robot could result in more rules being required in the controller. Kwak [59] has shown that large number of rules can compromise controller sample rates and degrade control performance. The aim in this thesis is to consider the design of a fuzzy controller from a real-time perspective in order to apply it to the Middlesex long reach robot. In order to achieve this aim it is desirable to have a design with the minimum amount of rules.

Table 2.1 and 2.2 summarise the attributes of the different controllers surveyed. The abbreviations used in the tables are listed below:-

TFMH	Transfer function modification through hardware.	P	Payload variation
TFMS	Transfer function modification through software.	V	Motion in the vertical plane
CRAM	Controllers requiring accurate models.	H	Motion in the horizontal plane
GC	Gravity Compensation.	AI	Artificial Intelligence.

Category	Controller	Structure	P	H	V	Reference
TFMH	adaptive	concentric beams		√	√	[28]
TFMH	classical (P)	micro-manipulator, external tip sensing		√		[30][33][32]
TFMH		concentric beams, external tip sensing		√		[69]
TFMH		micro-manipulator		√		[35]
TFMH		micro-manipulator, external tip sensing		√		[37]
TFMS	passive controller	reflected tip output, external tip sensing		√		[19][10][39][40]
CRAM	similar to input pre-shaping	slender theoretical structure, mass 0.0059 kg, external sensing		√		[45]
CRAM	open and closed loop input preshaping	slender theoretical structure		√		[42][43][41] [44]
CRAM	classical PD	two link flexible manipulator, external sensing		√	√	[70]

Table 2.1: Controllers and their properties - Part I

Category	Controller	Structure	P	H	V	Reference
CRAM	LQG	theoretical structure ignoring twist, external tip sensing		√		[9]
CRAM	Adaptive based on AR model	theoretical structure ignoring twist	√	√		[18][46]
CRAM	Adaptive STR	theoretical structure ignoring twist, external sensing	√	√		[47][50]
CRAM	Adaptive STR with online identification	theoretical structure ignoring twist, external sensing, dynamic payload	√	√		[49]
GC	PID	piezo electric hub actuators, external sensing			√	[54]
GC					√	[71][72]
GC	PID				√	[55]
AI	hierachical control with fuzzy supervisor				√	[56]
AI	fuzzy	slender theoretical structure			√	[60]
AI	fuzzy	slender theoretical structure	√		√	[63]
AI	neural network	slender theoretical structure	√		√	[64]
AI	fuzzy PD in TFS based configuration	1.18 m cylindrical cross section, internal tip sensing	√	√	√	[73][74][75][76] [77]

Table 2.2: Controllers and their properties - Part II

References

[1] Kuo B. C. *Automatic control systems - sixth edition.* Prentice Hall International, Inc., Englewood Cliffs, New Jersey 07632, 1991.

[2] Balas M. J. Feedback control of flexible systems. *IEEE Trans. on Automatic control*, AC-23(4):673–679, 1978.

[3] Balas M. J. Modal control of certain flexible systems. *SIAM J. Control and Optimisation*, 16(3):450–462, 1978.

[4] Truckenbrodt A. Truncation problems in the dynamics and control of flexible mechanical systems. In *Control Science and Technology for the progress of society. Proceedings of the eighth triennial world congress of IFAC*, pages 1909–1914, 1982.

[5] Benhabib R. J., Iwens R. P., and Jackson R. L. Stability of distributed control for large flexible structures using positivity concept. *J. Guidance and Control*, 4(5):487–494, 1981.

[6] Spector V. and Flashner H. Sensitivity of structural models for non-colocated control systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 111:646–655, 1989.

[7] Cannon R. H. and Rosenthal D. E. Experiments in control of flexible structures with non-collocated sensors and actuators. *J. Guidance and Control*, 7(5):546–553, 1984.

[8] Spector V. and Flashner H. Modelling and design implications of non-colocated control in flexible systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 112:186–193, 1990.

[9] Cannon R. H. and Schmitz E. Initial experiments on the end-point control of a flexible one-link robot. *The International Journal of Robotics Research*, 3(3):62–75, 1984.

[10] Pota H. R. and Vidyasagar M. Passivity of flexible beam transfer functions with modified outputs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2826–2831, 1991.

[11] Chodavarapu P. A and Spong M. On noncollocated control of a single flexible link. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1101–1106, 1996.

[12] Book W. J. Structural flexibility of motion systems in the space environment. *IEEE Transactions on robotics and automation*, 9(5):524–530, 1993.

[13] Kanoh H., Tzafestas S., Lee H. G., and Kalat J. Modelling and control of flexible robot arms. In *Proceedings of the* 25^{th} *IEEE CDC*, pages 1866–1870, 1986.

[14] Volkov D. P. and Zubkov Y. N. Vibrations in a drive with a harmonic gear transmission. *Russian Engineering Journal*, 58(5):11–15, 1978.

[15] Sweet L. M. and Good M. C. Redefinition of the robot motion control problem. In *IEEE control systems magazine*, pages 18–25, 1985.

[16] Kanoh H. and Lee H. G. Vibration control of a one link flexible arm. In *Proceedings of the* 24^{th} *IEEE CDC*, pages 1172–1177, 1985.

[17] Fukuda T. and Kuribayashi Y. Precise positioning and vibration control of flexible robotic arms with consideration of joint elasticity. In *Proceedings of IECON'84*, pages 410–415, 1984.

[18] Harashima F. and Ueshiba T. Adaptive control of a flexible arm using the end point position sensing. In *Proceedings of the Japan - U.S.A. symposium on Flexible Automation*, pages 225–229, 1986.

[19] Wang W. J., Lu S. S., and Hsu C. F. Experiments on the position control of a one-link flexible robot arm. *IEEE Transactions on Robotics and Automation*, 5(3):373–377, 1989.

[20] Dossing O. *Structural testing - Part II, Modal analysis and simulation.* Brüel & Kjaer, DK-2850 Naerum, Denmark., 1988.

[21] Luo Z. H. Direct strain feedback control of flexible robot arms : New theoretical and experimental results. *IEEE Transactions on Automatic control*, 38(11):1610–1622, 1993.

[22] Luo Z. H. and Sakawa Y. Direct strain feedback control of flexible robot arms via gain adaptation. In *Proceedings of IEEE TENCON'93*, 1993.

[23] Swevers J., Torfs F., Demeester F., and van Brussel H. Fast and accurate tracking control of a flexible one-link robot based on real-time link deflection measurements. *Mechatronics*, 2(1):29–41, 1992.

[24] Demeester F. and van Brussel H. Real-time optical measurement of robot structural deflections. *Mechatronics*, 1(1):73–86, 1991.

[25] Swevers J. An introduction to system identification and its application in practice. In De Schutter J. and Van Brussel, editors, *Computer controlled motion and robotics course*, pages 41–70. K.U. Leuven, Department of Mechanical Engineering, Leuven, Belgium, 1990.

[26] Gevarter W. B. Basic relations for control of flexible vehicles. *AIAA J.*, 8(4):666–672, 1970.

[27] Tarn T. J., Guo C., and Bejcy A.K. Issues in zeros of flexible robot systems. In *Proceedings of the* 32^{nd} *Annual Allerton Conference on Communication, Control and Computing*, pages 641–650, 1994.

[28] Zalucky A. and Hardt D. E. Active control of robot structure deflections. *Journal of Dynamic Systems Measurement and Control - Transactions of the ASME*, 106:63–69, 1984.

[29] Davis J. H. and Hirschorn R. M. Tracking control of a flexible robot link. *IEEE Transactions on Automatic control*, 33(3):238–248, 1988.

[30] Sharon A. and Hardt D. Enhancement of robot accuracy using endpoint feedback and a macro-micro manipulator system. In *Proceedings of the American Control Conference*, pages 1836–1842, San Diego Ca., 1984.

[31] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[32] Sharon A. Hogan N. and Hardt D. The macro/micro manipulator: an improved architecture for robot control. *Robotics and Computer-Integrated Manufacturing*, 10(3):209–222, 1993.

[33] Sharon A. Hogan N. and Hardt D. Controller design in the physical domain. *J. Franklin Inst.*, 328(5/6):697–721, 1991.

[34] Bekefi G. and Barret A. H. *Electromagnetic Vibrations Waves, and Radiation.* MIT Press, Cambridge MA, 1977.

[35] Chalhoub N. G. and Zhang X. Reduction of the end effector sensitivity to the structural deflections of a single flexible link: Theoretical and experimental results. *Journal of Dynamic Systems Measuremnt and Control, Transactions of the ASME*, 115(4):658–666, 1993.

[36] Lewis J. The design, construction and testing of a laser guided robot arm, MSc dissertation, Middlesex University, London, England. 1991.

[37] Chiang W. W., Kraft R., and Cannon R. H. Design and experimentaldemonstration of rapid precise end-point control of a wrist carried by a very flexible manipulator. *The International Journal of Robotics Research*, 10(1):30–40, 1991.

[38] Desoer and Vidyasagar. *Feedback systems: Input output properties.* Academic Press, NY, 1975.

[39] Wang D. and Vidyasagar M. Passive control of a single flexible link. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1432–1437, 1990.

[40] Wang D. and Vidyasagar. Passive control of a stiff flexible link. *International Journal of Robotics and Automation*, 11(6):572–578, 1992.

[41] Bhat S., Tanaka M., and Miu D. K. Experiments on point-to-point position control of a flexible beam using Laplace transform technique - Part I: Open loop. *ASME Journal of Dynamic Systems, Measurement and Control*, 113:432–437, 1991.

[42] Bhat S. and Miu D. K. Precise point-to-point positioning control of flexible structures. *ASME Journal of Dynamic Systems, Measurement and Control*, 112:667–674, 1990.

[43] Bhat S. and Miu D. K. Solutions to point-to-point control prblems using laplace transform techniques. *ASME Journal of Dynamic Systems, Measurement and Control*, 113:425–431, 1991.

[44] Bhat S. and Miu D.K. Experiments on point-to-point position control of a flexible beam using Laplace transform technique - Part II: Closed loop. *ASME Journal of Dynamic Systems, Measurement and Control*, 113:438–443, 1991.

[45] Grieco J. C., Fernandez G., Gamarra-Rosado V. O., and Armada M. Active compensation control of the flexural error of elastic robot manipulators. In *Preprints of the IFAC Workshop on 'Human-Oriented Design of Advanced Robotics Systems', DARS' 95, Vienna, Austria.*, pages 221–229, 1995.

[46] Harashima F., Nishiyama Y., Ueshiba T., and H. Hasimoto. Adaptive control of a flexible arm with a variable payload. In *Distributed Parameter Systems: Modelling and Simulation*, pages 323–328, 1989.

[47] Rovner D. M. and Cannon R. H. Jr. Experiments toward on-line identification and control of a very flexible one- link manipulator. *The International Journal of Robotics Research*, 6(3):3–19, 1987.

[48] Astrom K.and Borison U., Ljung L., and Wittenmark B. Theory and applications of self tuning regulators. *Automatica*, 13:457–476, 1977.

[49] Alder J. and Rock S. M. Experiments in control of a flexible-link robot manipulator with unknown payload dynamics:An adaptive approach. *The International Journal of Robotics Research*, 13(6):481–495, 1994.

[50] Rovner D. M. and Franklin G. F. Experiments in load adaptive control of a very flexible one-link manipulator. *Automatica*, 24(4):541–548, 1988.

[51] Tzes A. P. and Yurkovich S. Application and comparison of on-line identification methods for flexible manipulator control. *The International Journal of Robotics Research*, 10(5):515–527, 1991.

[52] Cetinkunt S. and Wu S. Output predictive adaptive control of a single link flexible arm. *International Journal of Control*, 53(2):311–333, 1991.

[53] Koivo A. J. Lee K. S. Self tuning control of a two link manipulator. *The International Journal of Robotics Research*, 11(4):383–395, 1992.

[54] Mulders P. C., van der Wolf A. C. H., Heuvelman C. J., and Snijder M. P. A robot arm with compensation for bendings. *Annals of CIRP*, 35(1):305–308, 1986.

[55] De Luca A. and Panzieri S. An iterative scheme for learning gravity compensation in flexible robot arms. *Automatica*, 30(6):993–1002, 1994.

[56] Garciabenitez E., Yurkovich S., and Passino K. Rule-Based supervisory control of a two link flexible manipulator. *Journal of Intelligent and Robotic Systems*, 7:195–213, 1993.

[57] Tzes A., Englehart M., and Yurkovich. Input preshaping with frequency domain information for flexible link manipulator control. In *Proc. AIAA Guidance, Navigation and Control Conference*, Boston, MA, 1989.

[58] Hillsley K. and Yurkovich S. Vibration control of a two-link flexible robot arm. In *Proc. IEEE Int.Conference on Robotics and Automation*, 1991.

[59] Kwak M. K. and Sciulli D. Fuzzy logic based vibration suppression control experiments on active structures. *Journal of Sound and Vibration*, 191(1):15–28, 1996.

[60] Moudgal V. G., Kwong W. A., Passino K. A., and Yurkovich S. Fuzzy learning control for a flexible link robot. *IEEE Transactions on Fuzzy Systems*, 3(2):199–210, 1995.

[61] Layne J. R. and Passino K. M. Fuzzy model reference learning control. In *Proc. 1st IEEE Conference on Control Applications*, pages 686–691, 1992.

[62] Procyk T. J. and Mamdani E. H. Linguistic self organising process controller. *Automatica*, 15:15–30, 1979.

[63] Wu J. and Tsuei Y. G. Comparison of fuzzy logic and self-tuning adaptive control of single-link flexible arm. *Mechatronics*, 3(4):451–464, 1993.

[64] Takahashi K. and Yamada I. Neural network based learning control of flexible mechanism with application to single-link flexible arm. *Transactions of the ASME*, 116:792–795, 1994.

[65] Kubica E. and Wang D. Fuzzy control strategy for flexible a single link robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 236–241, 1993.

[66] Lee J., Vukovich G., and Sasiadek J.Z. Fuzzy control of a flexible link manipulator. In *Proceedings American Control Conference, Baltimore, Maryland*, pages 568–574, 1994.

[67] Larkin I.L. A fuzzy logic controller for aircraft flight control. In Sugeno M., editor, *Industrial Applications of Fuzzy Control*, pages 87–104. 1985.

[68] Drainkov D., Hellendoor H., and Reinfrank M. *Introduction to Fuzzy Control.* Springer Verlag, 1993.

[69] Davies J. B. C. Elephants trunks an unforgettable alternative to rigid mechanics. *Industrial Robot*, 18:29–30, 1991.

[70] Chonan S. and Yamazaki M. Out of plane control of a two link flexible arm by tip sensing and base torquing. *Journal of Sound and Vibration*, 157(2):317–330, 1992.

[71] Jiang Z. H., Uchiyama M., and Hakomori K. Active compensating control of the flexural error of elastic robot manipulators. *Distributed Parameter Systems: Modelling and Simulation*, pages 369–373, 1989.

[72] Uchiyama M., Jiang Z. H., and Hakomori K. Compensating control of a flexible robot arm. *Journal of Robotics and Mechatronics*, 2(2):97–106, 1990.

[73] Surdhar J. S., White A. S, Stoker M., Gill R., and Korhonen J. A transputer implementation of a fuzzy PD controller applied to a flexible link manipulator. In *Proc. DARS'95 - Workshop on Human Oriented Design of Advanced Robotic Systems*, Vienna, Austria, 1995. ISBN 08 04 26042.

[74] Surdhar J. S., White A. S., Gill R., and Mistry G. Fuzzy PD control applied to a 1DOF flexible link manipulator in a novel tip sensor based configuration. In *Proc. of the International Workshop on Advanced Robotics and Intelligent Machines*, Salford, Manchester, 1996.

[75] Surdhar J. S., White A. S., Gill R., and Mistry G. An efficient fuzzy algorithm applied to a flexible link 1dof robot. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 68–72, New Orleans, Louisiana, USA, 1996.

[76] Surdhar J. S., White A. S., and Gill R. Neural network control applied to a flexible link manipulator in a tip feedback sensor based (TFS)based configuration. In *Proc. of the* 12^{th} *International Conference on CAD/CAM Robotics and Factories of the Future*, London, UK, 1996.

[77] Surdhar J. S. and Korhonen J. A fuzzy PD controller applied to a long reach manipulator. In *Middlesex University Faculty of Technology Technical Report Series ,MUCORT - Middlesex University Research Conference*, Middlesex University, London, 1995. ISSN 1362-2285.

Chapter 3

Modelling the TFS based flexible robot

3.1 Introduction

Chapter 2 provides a review of methods other workers investigated in order to achieve a controllable minimum-phase model. Generally the main methods include altering the transfer function of the system physically through using end-point actuation devices such as micro-manipulators or producing stable controllers based on accurate modelling.

The control solution proposed in this thesis does not rely on a model hence the problem of obtaining a non-minimum phase model or a highly accurate model is avoided. However, in order to synthesise a controller and verify its stability a model is required. It will also be a useful investigation for future workers who may wish to investigate a model based approach.

This chapter outlines the development and validation of a mathematical model used to simulate the behaviour of a manipulator with a single flexible link. The model is based on a numerical solution, using the finite difference technique, of a Lagrange-Euler formulation of a continuous shaft which takes in account bending and shear moments [1]. The model of a single flexible link has been previously validated and compares well (within 3%) to an analytic solution [2].

The model has been extended to include feedback from the tip of the arm via the tip based sensor which has a non-linear transfer function. Further non-linear effects due to the motor current saturation limits have been included. Validation of the extended model for the first fundamental mode has been carried out using empirically obtained frequency data and in the time domain. Responses from higher vibration modes are filtered out in the practical implementation of the controller in order to prevent observer spillover (see Chapter 6). The model is used to show the stability of a simple classical control scheme in the TFS based configuration. Root locus and transient behaviour of a PD controller is investigated as this controller forms the basis of the fuzzy controller proposed in the next chapter of this thesis.

The model has been programmed in Advanced Continuous Simulation Language (ACSL) [3]. ACSL is a programming environment based on FORTRAN that has been designed for modelling and evaluating the performance of continuous systems described by time-dependent, non-linear differential equations.

This chapter is divided into the following stages which are the usual procedure for the development of a mathematical model of a dynamic system [4]:-

Physical modelling This stage involves specification of a physical model of the system which captures the salient features of the system being modelled. For the purpose of this thesis, the prototype system developed by Lewis [5] is the system to being modelled.

Mathematical modelling This usually involves derivation of differential equations of motion that describe the dynamic behaviour of the system.

Mathematical model investigation Here the derived equations are solved subject to the given constraints such as boundary or initial conditions and system parameters.

Model validation This is an ongoing process with the purpose of developing confidence in the completeness, precision and accuracy of the model over the required range.

Analysis The validated model is then used to study the effects of the variation of different parameters on the system. These are to test the robustness of the model and to gain insight into the behaviour of the physical system.

3.2 Physical modelling

This stage constitutes of establishing an 'ideal' model of the real system. Assumptions are made in order to simplify the modelling and analysis procedure and boundary conditions are specified to represent interactions between the real system and its environment. Bending and shear moments of the arm are also considered.

Torsion effects have not been included in the simulations but can be by addition of an extra shear term (other methods such as those based on the separation of variables cannot be modified to account for torsion effects). In this context the model resembles the 'theoretical' structures reviewed in the previous chapter. A 'theoretical' structure is defined as a slender beam, usually used in experimental studies that ignore the effects of torsion, that has a relatively large width and length in comparison to thickness of the beam.

A distributed model of a single flexible link has been chosen as the 'ideal' physical model of the system. By use of a distributed model as opposed to a lumped model a more accurate representation of the system can be obtained especially in the case where the end mass is not large enough to consider the rest of the arm without mass. The effects of the actuator, a DC servo motor, are also included.

Assumptions made in order to simplify the model are:-

- The effects of gravity are ignored,
- the effects of noise are neglected,
- and the system is not altered by the surrounding environment.

3.3 Mathematical modelling

This section describes the derivation of the dynamic equations of the arm, motor and tip sensor. The equations for the arm and motor are evolved using standard Lagrange and armature motor equations respectively. The model for the sensor is determined by mathematical approximation of empirically obtained data.

3.3.1 Equations of motion for the arm

This section will describe the derivation of the equations of motion for the arm structure. First equations for a segment of a the arm are derived. These equations are then extended to give the equations of motion for the arm tip. The method of numerical solution of these equations using the finite difference approach is then outlined.

Equations of motion for a segment of the beam

For this analysis, it is assumed that the beam is small in diameter compared to it's length. This implies that sections remain plane and translate but do not rotate, i.e., rotary inertial forces are ignored. Consider small displacements of the robot arm as shown in Fig. 3.1 where ρ, E and I are

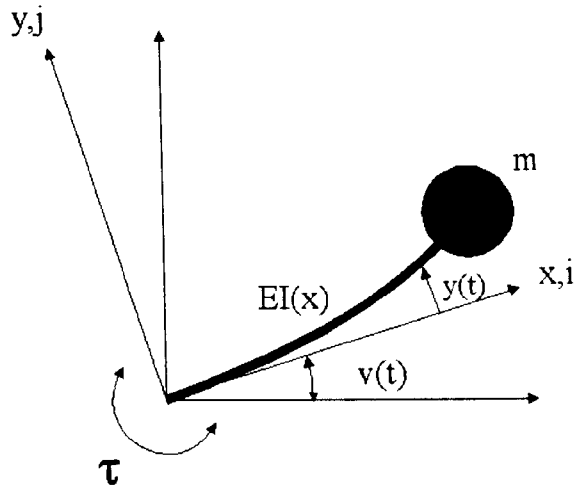

Figure 3.1: Deflection of the robot arm

all functions of the arm or link length, x. Let a small segment of the arm be at a distance x from the origin. This is displaced by $y(x,t)$. The position vector is given by :-

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} \tag{3.1}$$

If the co-ordinate system is rotating at $\omega(t) = d\theta/dt$, the velocity of the segment in inertial space will be given by:-

$$\mathbf{v} = \frac{\partial \mathbf{r}}{\partial t} + \omega \times \mathbf{r} \tag{3.2}$$

$$= \dot{x}\mathbf{i} + \dot{y}\mathbf{j} - \mathbf{i}y\omega + \mathbf{j}x\omega \tag{3.3}$$

For small movements second order terms can be ignored so that $y\omega$ cancel out and

$$\frac{dx}{dt} \approx 0 \tag{3.4}$$

$$\mathbf{v} = (\dot{y} + x\omega)\mathbf{j} \tag{3.5}$$

$$\dot{\mathbf{v}} \approx (\ddot{y} + x\dot{\omega})\mathbf{j} \tag{3.6}$$

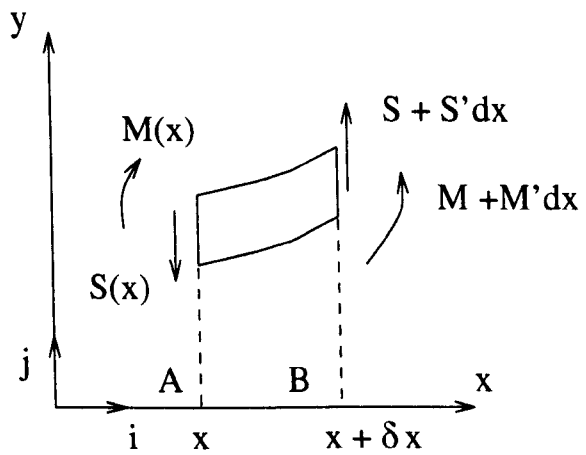

Figure 3.2: Elementary portion of a uniform beam

Considering the forces on an elementary portion of a beam segment as shown in Fig. 3.2, the mass of the segment is ρdx. At equilibrium the forces acting on the segment are:-

$$\rho dx\dot{v} = (S + \frac{\partial S}{\partial x}dx - S)\mathbf{j}$$

$$= \frac{\partial S}{\partial x} dx\mathbf{j} \tag{3.7}$$

$$\tag{3.8}$$

or

$$\frac{\partial S}{\partial x} = \rho(\ddot{y} + x\dot{\omega}) \tag{3.9}$$

Taking moments about side B results in

$$\begin{aligned} Sdx + (M + \frac{\partial M}{\partial x} dx) - M &= 0 \\ \Rightarrow \qquad S &= -\frac{\partial M}{\partial x} \end{aligned} \tag{3.10}$$

using

$$M = EI \frac{\partial^2 y}{\partial x^2} \tag{3.11}$$

from Euler's theory of bending and eqs. 3.9 and 3.10 we obtain:

$$\rho(\ddot{y} + x\dot{\omega}) = (EI \frac{\partial^4 x}{\partial x^4})(x,t) \tag{3.12}$$

Rewriting 3.12 gives equation of motion for a segment of the arm:-

$$\ddot{y} + \frac{1}{\rho}(EI \frac{\partial^4 x}{\partial x^4})(x,t) = -x\dot{\omega} \tag{3.13}$$

Equation of motion for a mass at the arm's tip

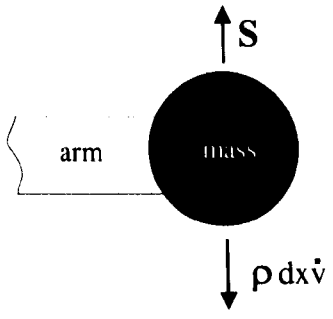

Figure 3.3: Forces acting at the tip of the arm

from fig. 3.3

$$m\dot{\mathbf{v}} = -S(L,t)\mathbf{j} \tag{3.14}$$

using Eq.3.13, Euler's theory of bending and Eq. 3.10 the equation of motion for a mass at the tip is derived as:

$$\frac{m}{\rho} \frac{\partial^4 y}{\partial x^4}(L,t) + \frac{\partial^3 y}{\partial x^3}(L,t) = 0 \tag{3.15}$$

The model solution is subject to the following boundary conditions:-
The arm is fixed at the root:-

$$y(0,t) = \frac{\partial y}{\partial x}(0,t) = 0 \tag{3.16}$$

assuming the arm starts from rest the initial conditions are:-

$$\frac{\partial^2 y}{\partial x^2}(L,t) = 0 \tag{3.17}$$

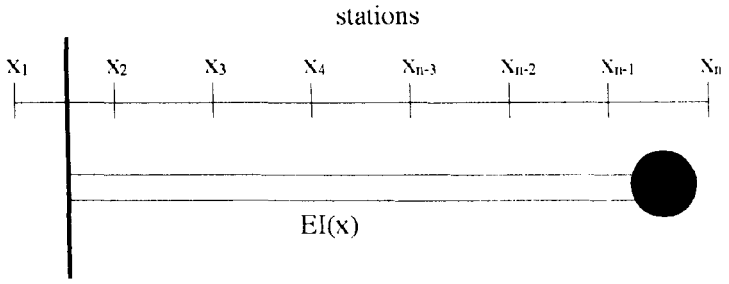

Figure 3.4: Finite difference segments

Solution of the arm equations by finite difference techniques

For numerical solution using Advanced Continuous Simulation Language (ACSL) the *central-difference* finite difference approximations and an explicit[1] method of solution is used [6]:-

$$y_n = \frac{y_{n+1} + y_{n-1}}{2}$$

$$\frac{\partial y}{\partial x}|_n = \frac{y_{n+1} - y_{n-1}}{2\Delta x}$$

$$\frac{\partial^2 y}{\partial x^2}|_n = \frac{y_{n+1} + y_{n-1} - 2y_n}{(\Delta x)^2} \tag{3.18}$$

The arm is divided into n stations, δx, as shown in Fig. 3.4. In order to represent the boundary condition derivative $\partial y/\partial x$ more accurately at x=0 and x=L, whilst using the central-difference approximation, it is necessary to introduce 'fictitious' displacements, corresponding to y_{n-1} and y_{n+1}, that satisfy the boundary conditions, at the external mesh points by extending the arm slightly in a negative and positive directions.

The mesh length δx is given by:

$$\delta x = \frac{L}{n-2} \tag{3.19}$$

Where L and n are the length of the arm and number of stations respectively. The following boundary conditions are used:

$$y_1 = y_2 = 0$$

$$M_{n-0.5} = 0; \quad \frac{M_n - M_{n-1}}{2} = 0$$

$$\frac{\partial^3 y}{\partial x^3}(L,t) = \frac{\partial M}{\partial x}|_{n-0.5} = \frac{M_n - M_{n-1}}{dx} \tag{3.20}$$

Using a Taylor expansion the fictitious stations are eliminated and the following auxiliary quantity is obtained:-

$$\frac{3h^2}{2}\frac{\partial^2 M}{\partial x^2}|_{n-0.5} = 3M_n + M_{n-2} \tag{3.21}$$

Substituting eq. 3.21 into 3.15 and using the boundary conditions in 3.20, the following solution is obtained:-

$$M_n\left[1 + \frac{M}{\rho Dx}\right] = -M_{n-2}\left[\frac{M}{3\rho Dx}\right] \tag{3.22}$$

[1] a formula which expresses one unknown pivotal value directly in terms of known pivotal values

It should be noted that the explicit solution using the central difference approximation is stable for small integration time steps satisfying [6]:-

$$0 < \frac{\delta t}{(\delta x)^2} \leq 0.5 \tag{3.23}$$

Eq. 3.23 implies that small integration time steps are required in the computer simulation. In the simulation this constraint is easily satisfied since $\delta x = 0.265$ and $\delta t = 0.0001$.

3.3.2 Motor and drive characteristics

The motor characteristics are included in the arm dynamics by replacing $\dot{\omega}$ in Eq. 3.13 with a disturbance at the root given by the standard equations of motion of an armature controlled motor [7][8]. Eq. 3.24 gives the standard motor equations including the arm dynamics with the torque referred to the output shaft.

$$\dot{\omega}_L = \frac{\tau}{NJ_e} - \frac{\omega\mu_e}{J_e} + \frac{EI}{N^2 J_e}\frac{\partial^2 M}{\partial x^2} \tag{3.24}$$

where EI is the stiffness coefficient of the flexible arm, $J_e = (J_m + J_l/N^2)$ and $\mu_e = (B_m + B_l/N^2)$ are the effective moment of inertia and friction (viscous) of the system.

Fig. 3.5 shows a simple circuit model of the armature control system.

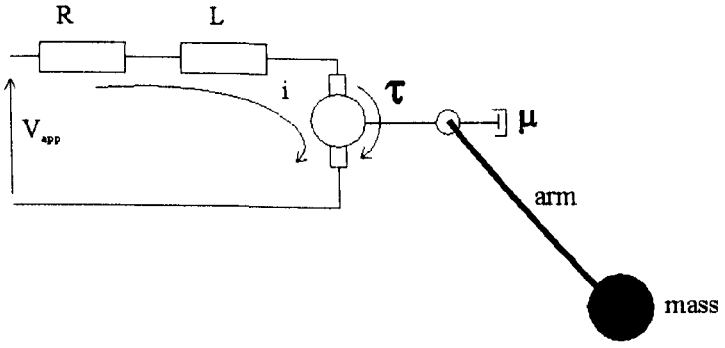

Figure 3.5: Armature control of the servo motor

Applying Kirchhoff's [9] voltage law yields

$$V_{app} = Ri + L\frac{di}{dt} + \omega(t)K_E \tag{3.25}$$

here K_E represents the back EMF constant of the motor, R and L the armature resistance and inductance, V and I the armature voltage and current respectively. Assuming the torque generated by the armature moving in a permanent magnetic field is linearly related to the current [10]

$$\tau = K_T I \tag{3.26}$$

where K_T is the torque constant.

Closing the loop about the and with reference to the commanded input, θ_c yields the error to be minimised, e.

$$e = \theta_c - \theta \tag{3.27}$$

The control input, u, under Proportional control is

$$u = K_p.e \tag{3.28}$$

in the Eq. 3.28 above K_p is the proportional gain constant.

Fig. 3.6 depicts a simple closed loop block diagram for the hub-based control system.

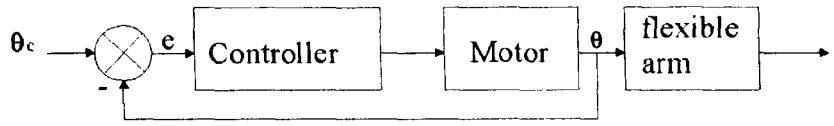

Figure 3.6: Block diagram of the actuator and flexible arm

3.3.3 Sensor equations

This section will describe the modelling of the sensor and its inclusion in the complete model including the motor and arm. The model of the sensor is derived from empirical data. This model is then added to the model of the motor and the arm to incorporate the tip measurement.

Measured sensor characteristics

Fig. 3.7 below depicts the experimentally obtained sensor transfer function by Lewis [5]. The non-linear transfer function for the optical sensor is obtained by drawing the laser across the diameter of the sensor and measuring its output. The output from the sensor is a voltage corresponding to the deflection of the arm from the desired position defined by the optical head or laser.

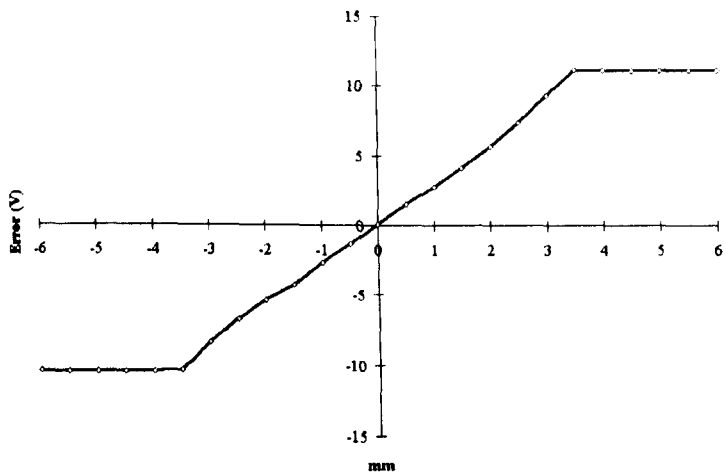

Figure 3.7: Non-linear transfer function for the optical sensor

Section 3.6.1 shows that the gradient of the saturating function can be varied depending on the size of the impinging light spot. For the purposes of modelling, and in the experimental observations carried out in this dissertation, this gradient is kept constant. A simple saturating function can be used to model the non-linear transfer function.

Inclusion of the tip feedback parameter

The arm deflection as computed by the finite difference solution above is given by:-

$$\Delta\theta = 1.5 * y_{n-1} - 0.5 * y_{n-2} \tag{3.29}$$

The total displacement of the tip of the arm will therefore be the sum of the rigid body deformation, given by the hub angle, plus the tip deflection. This deflection computation is transformed by the non-linear saturating function f_h, which models the sensor, before its inclusion in the model. Eq. 3.30 describes this relationship and the error quantity, e, in Eq. 3.27 is redefined as:-

$$e = \theta_c - (\theta + f_h \Delta\theta) \tag{3.30}$$

The sensor maps the actual position error signal in X-Y space to a corresponding error voltage signal in the X (horizontal) and Y (vertical) plane. The X and Y error voltages are used to control the corresponding individual axes of the robot.

The model is further simplified based on observations of the tracking performance of the robot under normal operating conditions. Fig. 3.8 depicts the tracking performance of the end-effector with a 1.6 kg payload[2]. The arm was commanded to move 10° in the horizontal plane. A classical PD controller was used for this experiment.

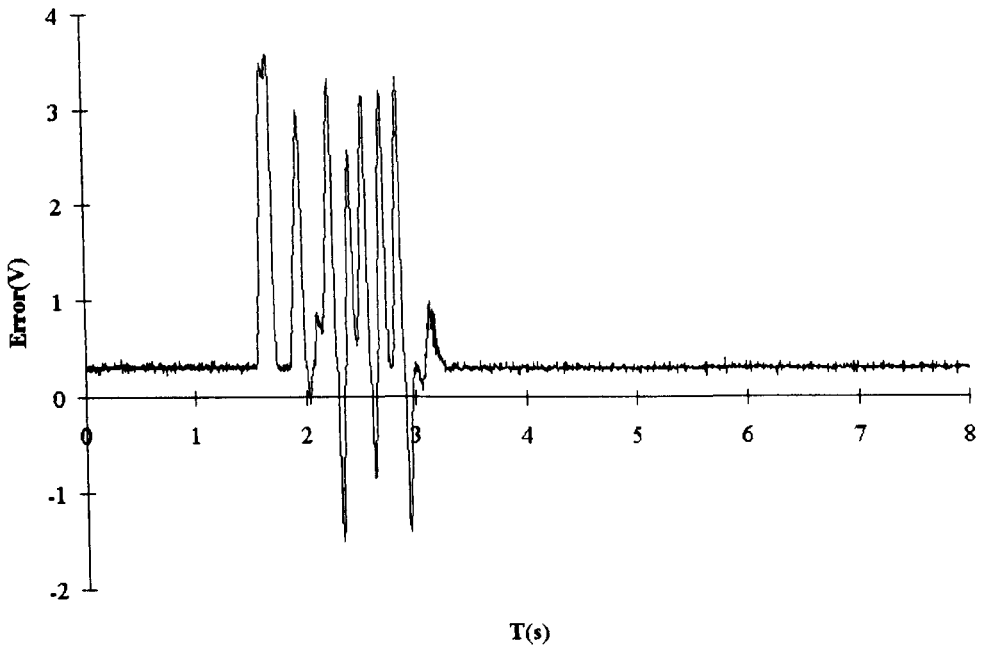

Figure 3.8: End-effector tracking performance

The main observation from the figure above is that the sensor does not saturate under normal operation for the maximum load condition. Sensor saturation would be observed if the output voltage of the sensor were to exceed 10 V. This implies that the function f_h in Eq.3.30 can be approximated by a linear function. It should be also noted that the magnitude of the largest deflection measured is 3.63 volts i.e., $\approx$ 36% of the signal before saturation. This would be the logical step height for a transient response analysis. However, in the simulation and practical experiments a higher step height is used (corresponding to a value just before the saturation of the sensor). The reasons for this choice are:-

- The 0.003 radian step height provides an easy reference whilst performing the practical experiments, i.e., the step height is set at the point where the sensor will begin to saturate,
- The 0.003 radian step is larger than the system would normally experience under normal tracking operation ($\approx$ 0.00108 radians is the largest deflection measured for tracking). Applying the larger step adds an element of robustness testing for the non-linear model.

K_{TIP} a tip feedback parameter, the gradient of f_h, has been included as a conversion constant between the measured tip position (error) and its corresponding voltage which is negatively fed back into the controller via a summing junction. The value for K_{TIP} was kept constant at 3.3 V/mm both in the simulation and practical experiments. This corresponds to a spot size of 7.5 mm which is three quarters the diameter of the quadrant detector. For simulation purposes K_{TIP} is normalised to 1 by applying a step input of magnitude 3 mm (0.0031 radians for a 1.18 m arm) which is the maximum step size than can be applied to the system in practice before saturation. The block diagram for the revised system with tip feedback is depicted in Fig. 3.9 below.

The approach of modelling a flexible system with non-collocated sensors via a transformed or *modified output* measurement has been investigated by other workers including Chodavarapu and Spong [11] and the workers from the Waterloo University group including Wang [12], Pota [13] and Vidyasagar [14]. These workers investigated modified output measurements, described in Eqs.

[2]this is the maximum payload applied to the robot in all the experiments

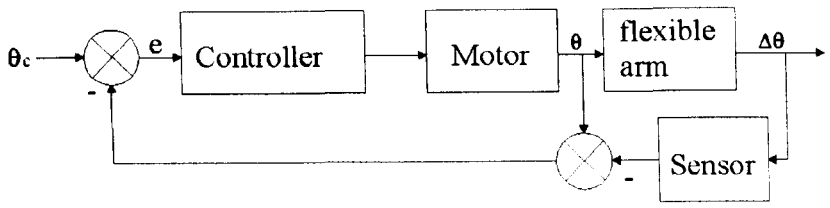

Figure 3.9: Block diagram of the actuator and flexible arm with tip feedback incorporated

3.31, 3.32, 3.33 and 3.34, in a bid to achieve a minimum phase model which would in turn yield a stable passive controller.

(1) Arclength of the tip:-

$$y_1 = h(q) = L\theta(t) + w(L,t) \tag{3.31}$$

(2) Reflected tip position:-

$$y_2 = h(q) = L\theta(t) - w(L,t) \tag{3.32}$$

(3) Reflected tip position with weighting:-

$$y_3 = h(q) = L\theta(t) - \kappa w(L,t) \tag{3.33}$$

(4) Virtual angle of rotation:-

$$y_4 = \theta(t) + \kappa \frac{\partial w}{\partial x}(L,t) \tag{3.34}$$

In the equations above $\theta(t)$ is the hub angle and $w(L,t)$ is the transverse displacement of the tip. The salient points from the literature by the workers mentioned above are re-iterated here.

Reflected tip position Pota and Vidyasagar [13] and Wang and Vidyasagar [14] studied systems with a *reflected* tip position, i.e., one where the tip position is given by the rigid body deformation *minus* the tip deformation. With such an output measurement it was possible to achieve a minimum phase transfer function provided the hub the ratio of the hub inertia to the arm inertia was sufficiently large (or very small in the case of a uniform beam). They showed also that a *weighted* reflected output measurement yielded a minimum phase transfer function that was controllable by a passive PD controller. In the latter case, a high ratio of hub to arm inertia was not important for stability.

Stability issues Tarn *et al.* [15] analysed a range of modified end point output measurements, using a non-linear infinite dimensional model, including the arc length of the tip, the reflected tip position and reflected tip position with weighted vibration. They found them all to have RHS zero dynamics, i.e., non-minimum phase, and therefore unstable under passive control. However, their analysis for the system zero dynamics relied on an infinite dimensional model without damping. Although an infinite dimensional model is supposed to be more accurate, it would be inaccurate in a practical system, even for the first mode, if the effects of damping in the beam were ignored. In the Waterloo experiments it proved to be virtually impossible to excite the 4^{th} mode with a pure sinusoid, hence use of a finite second order model was a valid approach. The work in this thesis differs from Tarn *et al.* and the Waterloo group in that a 16^{th} order model is used and damping is introduced into the system via the actuator dynamics.

The virtual angle measurement Chodavarapu and Spong investigated the *virtual angle* modified output. This measurement consisted of the rigid deformation plus a weighted value of the slope of the beam at its tip. It was shown that this measurement yielded a system which had poles and zeros lying alternately on the imaginary and the phase lag due to the flexible modes never exceeding 180°. Although this minimum phase system was also able to satisfy

the analysis of Tarn *et al.* it is clear that by using the measurement of the slope of the tip the advantages of absolute tip positioning are lost. Accurate placement of the tip cannot be guaranteed and the scheme will not be able to compensate for a permanent deformation of the link as only a slope sensor as opposed to a position sensor is used.

From the outputs described above it becomes immediately apparent that measurement via the optical sensor proposed in this thesis is similar to the weighted reflected tip output measurement of Pota and Vidyasagar [13]. In the case of TFS based control the tip feedback parameter K_{TIP} may be seen to perform the same role as the output weighting, κ, as described in Eqs. 3.33 and 3.34. Section 3.6.2 will analyse the zero dynamics of the model using the root locus technique in a bid to discern whether it yields a minimum phase system that can be controlled by a passive controller such as PD. Both arclength of the tip and reflected output measurements are investigated.

3.4 Model parameters

Identification of the model parameters is described in this section. The aim of identification is to determine the model parameters for the actuator, tip sensor and the arm structure and to include them into the model. Fig. 3.10 depicts the co-ordinate system used to describe the parameter characterisation. This co-ordinate system is used universally throughout the thesis.

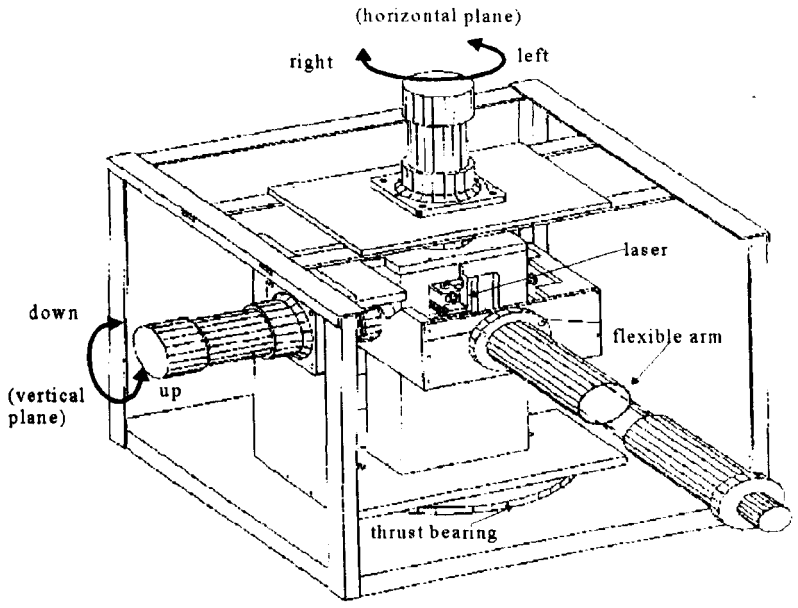

Figure 3.10: Co-ordinate system for experiments carried out on the TFS based robot.

3.4.1 Motor and drive parameters

Appendix E lists the motor characteristics for combination of the DC motor and Harmonic Drive gears used. These parameters include the 100:1 gear ratio that the motors were supplied with[3]. The gear ratio was later changed to a value of 1000:1 for the following reasons:-

- Finer movement at the tip.
- Greater stability when controlling the manipulator with a flexible link as the effect of the disturbance of the flexible arm is greatly reduced in comparison to the dynamics of the motor [11].
- Safety whilst tests were being carried out on the practical system - the effect of the higher gearing decreases the angular velocity of the arm thus reducing the chances of damage to operator and the system itself.

[3]The moment of inertia is referred to the output shaft resulting from the sum of the motor inertia and Harmonic Drive inertia

Current saturation limits of ± 1.9 A are used in the dynamic equations of the motor. These are the rated values as opposed to peak limits. A gear efficiency, η, of 60%, has also been included in the model as specified in the datasheets.

3.4.2 Structural parameters

The 'structure' comprises of the parameters associated with the flexible aluminium link. The table 3.1 summarises the measured structural parameters [16].

Parameter	Description	Value	Unit (S.I)
EI_m	Measured product of Young's modulus and 2^{nd} moment of inertia	8890	Nm^2
EI_c	Calculated product of Young's modulus and 2^{nd} moment of inertia	8929	Nm^2
ρ	Mass per unit length	0.75	kg/m
m	Mass of tip sensor	0.15	kg
L	Length of beam	1.18	m

Table 3.1: Structural parameters

EI, the product of Young's modulus and the 2^{nd} moment of inertia of the arm, was inferentially determined through measurements of the unloaded and loaded arm deflections and by use of Euler's beam theory. The value given is an average taken over the different payloads applied for the deflection measurements (varying between 0 - 2 kg). This measurement is in agreement with the theoretical determination using Young's modulus for aluminium. Although the measurements and theory are compatible there is a small difference in the values. This could arise due to a couple of factors:-

- An imperfect arm support for the clamped measurements,
- Non-uniformity in the thickness of the arm (error ± 0.000001 m).[4]

3.4.3 Dynamic parameters of the arm

Measurements of the moments of inertia, J_{arm}, and viscous friction, μ_{arm}, of the robot arm for each axis of rotation are summarised in the tables below. Determination of these values was carried out by considering the equation of motion for a transient vibration and solved through measurements of the parameters associated with the natural damping of the arm [16]. The experimental arrangement included complete construction of the robot, facilitating the measurements to be influenced by factors such as the viscous friction of the bearings and actuator gearing. Table 3.2 gives the parameters associated with movement in the vertical plane i.e., about the horizontal axis. Fig. 3.11 shows the experimental arrangement used to determine the parameter values.

Parameter	Description	Value	Unit (S.I)
J_h	Moment of inertia, horizontal axis	0.22 ± 0.005	kgm^2
μ_h	Viscous friction, horizontal axis	0.127± 0.013	kgm^2s^{-1}

Table 3.2: Dynamic parameters for rotation about the horizontal axis of rotation

For movements of the arm in the horizontal plane a series of measurements were made based on different elevations, $\delta°$, of the arm (cf. the experimental arrangement in Fig. 3.12). The values given are for the unloaded case. Pape [16] observed that the influence of the moments of inertia due to the additional weights was negligible in comparison to the moment of inertia of the robot construction. Table 3.3 summarises the measurements.

[4] outer diameter = 0.0000634 m and the inner diameter = 0.0000608 m.

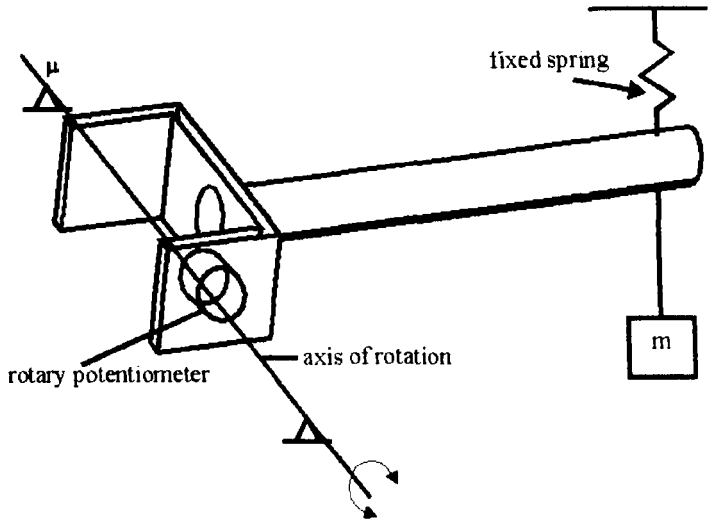

Figure 3.11: Experimental arrangement for measurements in the vertical plane

Parameter	Description	Value	Elevation	Unit (S.I)
J_{arm}	Moment of inertia, vertical axis	0.639 ± 0.008	0°	kgm^2
J_{arm}	Moment of inertia, vertical axis	0.624 ± 0.008	9.8°	kgm^2
J_{arm}	Moment of inertia, vertical axis	0.596 ± 0.008	19.6°	kgm^2
J_{arm}	Moment of inertia, vertical axis	0.349 ± 0.004	29°	kgm^2
J_{arm}	Moment of inertia, vertical axis	0.271 ± 0.003	40.7°	kgm^2
μ_{arm}	Viscous friction, vertical axis	0.222 ± 0.004	0°	kgm^2s^{-1}
μ_{arm}	Viscous friction, vertical axis	0.218 ± 0.004	9.8°	kgm^2s^{-1}
μ_{arm}	Viscous friction, vertical axis	0.212 ± 0.004	19.6°	kgm^2s^{-1}
μ_{arm}	Viscous friction, vertical axis	0.155 ± 0.002	29°	kgm^2s^{-1}
μ_{arm}	Viscous friction, vertical axis	0.125 ± 0.002	40.7°	kgm^2s^{-1}

Table 3.3: Dynamic parameters for rotation about the vertical axis of rotation

Table 3.3 shows that there is a variation of up to 57.6% in the moment of inertia parameter and a variation of up to 43.5% in the viscous friction parameter for a variation in arm elevation of 0° - 40°. The ACSL simulation used the parameters for 0° elevation.

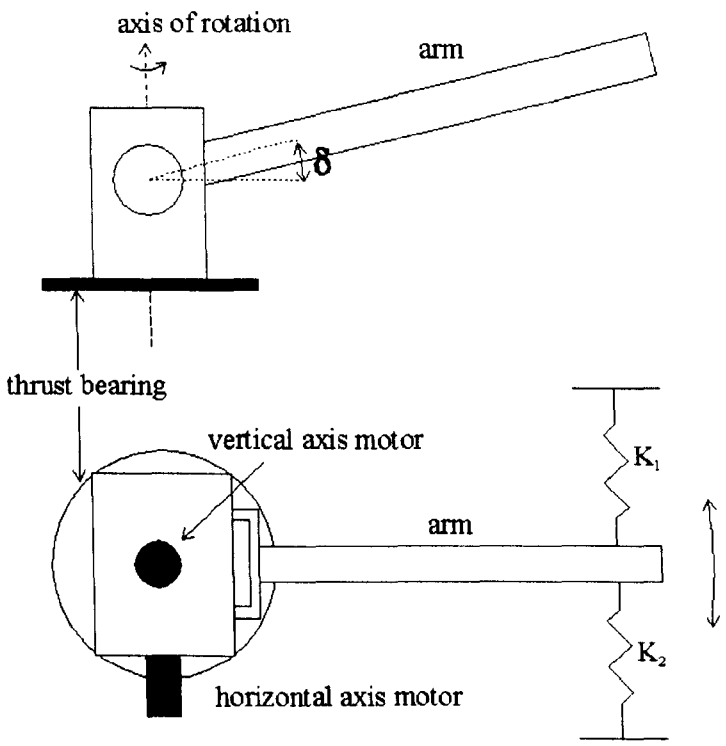

Figure 3.12: Experimental arrangement for measurements in the horizontal plane (K1 and K2 are spring constants used in experiments).

3.5 Model validation

This section will describe the validation of the model. Validation is carried out by comparing the frequency and transient responses of the model to those of the physical model.

3.5.1 Frequency response

The open loop frequency response of the physical system, composed of the actuator and flexible arm, has been measured using three different methods as described below. All these measurements have been carried for movement about the horizontal axis of rotation (up and down movement) :-

Impact hammer (IH) In this test the arm was excited by an impact to the tip and the frequency response was obtained using the Bruel & Kjaer real-time analyser. The frequency response in this case is mainly due to the arm, the link and root joint bracket.

FRA1 A Frequency response analyser (Schlumberger Solartron 1172) was used to monitor the acceleration at the tip of the arm via a tip mounted accelerometer (in the direction of excitation) and charge amplifier. Sinusoidal excitation of the horizontal axis actuator was carried out using the output from the Solartron analyser. Driver software written in-house by Stefan Okrongli[5] enabled the FRA to be interfaced directly to a PC for data capture and analysis using MS Excel spreadsheet software. The response in this case includes the dynamics of the arm, the link and bracket joints, backlash and flexure in the Harmonic Drive.

FRA2 In this experiment the tip of the arm was excited by a electromagnetic vibrator instead of the hub based sinusoidal excitation. A tip based accelerometer as in FRA 1 and a force transducer was used to monitor the tip vibrations via the Schlumberger Solartron analyser. The measured response includes the dynamics of the arm and the link and bracket joints and gear backlash.

The frequency responses using the different methods, IH, FRA1 and FRA2 of measurement are shown in Figs. 3.14, 3.15 and 3.16 respectively. The first fundamental modes are clearly identified on each graph. Modes are identified as the peak frequency that shows a 90° change in phase lag with the input signal.

It can be seen that each method yields a different fundamental mode because of the change in experimental conditions. Methods IH and FRA2 yield similar results, a fundamental mode near 16 Hz and a second and third mode at 136 Hz and 204 Hz respectively. A stroboscope was used to confirm these modes.

The FRA1 method yielded a slightly lower fundamental frequency of 14 Hz. A second anti-resonant mode is observed near 19 Hz. The lower fundamental frequency in comparison to the IH and FRA2 methods is attributed to the flexure in the Harmonic Drive and link and root joint bracket. The second anti resonant mode is possibly due to torsion effects caused by the (twisting moment produced by the actuator) in the structure as the other two methods, which tend to be more directional (the excitation from the hammer impact and vibrator were applied along the axis that bisects the length of the arm), do not display this behaviour. Fig. 3.13 illustrates these differences. It is not possible to confirm the second transversal mode using FRA1 as observed using the IH and FRA2 methods because at this frequency the amplitude is heavily attenuated (less than -40 dB) and not easily discerned from noise.

Responses for a loaded arm, with 0.45, 0.9 and 1.36 kg, using the FRA1 method were also taken. Fig. 3.17 depicts these results. In these measurements the fundamental mode is not easily discerned.

It can also be seen that any higher modes which may have shifted to lower frequencies due to loading are heavily damped in the region of 0 to 19 Hz. Beyond this region it can be seen that there exist a series of resonant peaks from higher modes that have shifted to lower frequencies due to loading. An example is the mode at 37 Hz in Fig. 3.16 which has shifted to 22 Hz in Fig. 3.17.

[5]a former Middlesex University research student

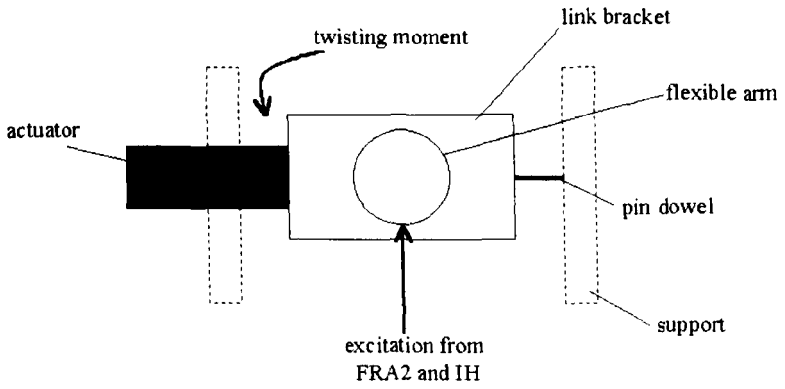

Figure 3.13: Front view of robot showing application of vibration excitation using the FRA1, FRA2 and IH methods.

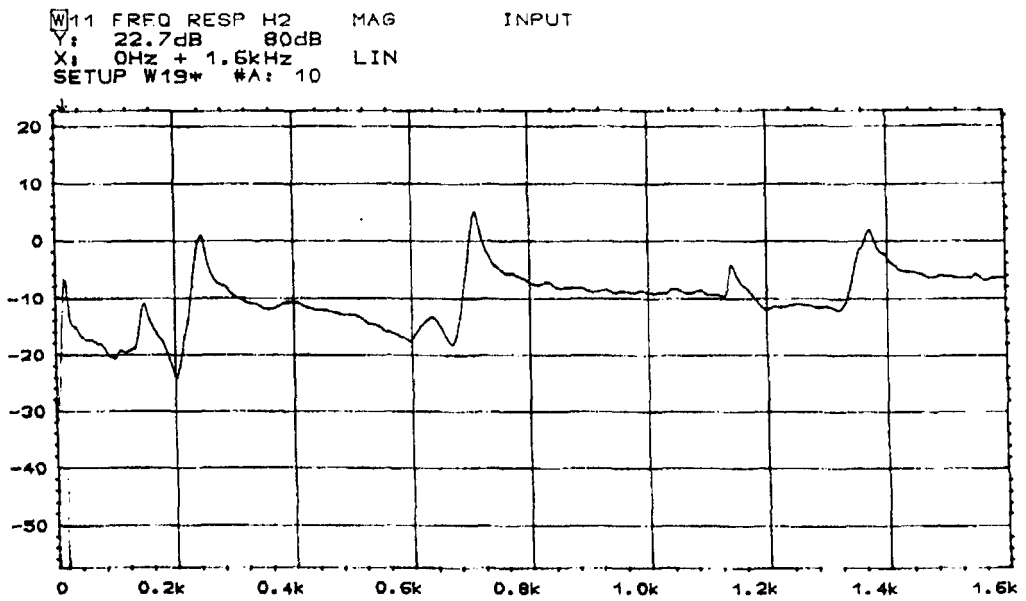

Figure 3.14: Measured frequency response data using the IH method

Simulated frequency response - arclength of the tip

Initial frequency plots for the simulation, with the parameters identified in sections 3.4.2 and 3.4.3, yielded a fundamental mode frequency of $\approx$ 31 Hz. This value is 17 Hz higher than the frequency measured for the same mode. The reason for this discrepancy is that the model did not account for the additional flexibility in the structure due to a flexible joint with Harmonic Drive gearing and a link and root joint bracket as shown in Figs 3.18 and 3.19 (The structural and dynamic parameters were determined using a clamped beam).

The model was adjusted to take in account the additional flexibility by taking advantage of the distributed nature of the model and incorporating a less stiff section between stations 1 and 2, $EI_{ROOT} = 4001$, in equation 3.13. The first station is also heavier than the rest of the arm so the mass per unit length, ρ, was also adjusted accordingly. The value of this combined joint and gear flexibility was estimated by comparing the simulated response to the measured response.

The resulting simulated open loop frequency response (including the DC motor and the arm) showed a dominant resonant mode at 14 Hz and an anti-resonant frequency at 19 Hz which is in agreement with the FRA1 measurements described earlier in this section. The simulated frequency response is depicted below in Fig. 3.20. It can be seen that the higher modes depicted in Figs 3.16 and 3.17 are completely damped. Since the controller design will be based on this reduced order model it is important to filter off higher un-modelled modes in the practical implementation to prevent the effects of spill-over [17]. This filter would be particularly important in the case of a model based controller. In the practical implementation a first order low pass filter with a cut-off

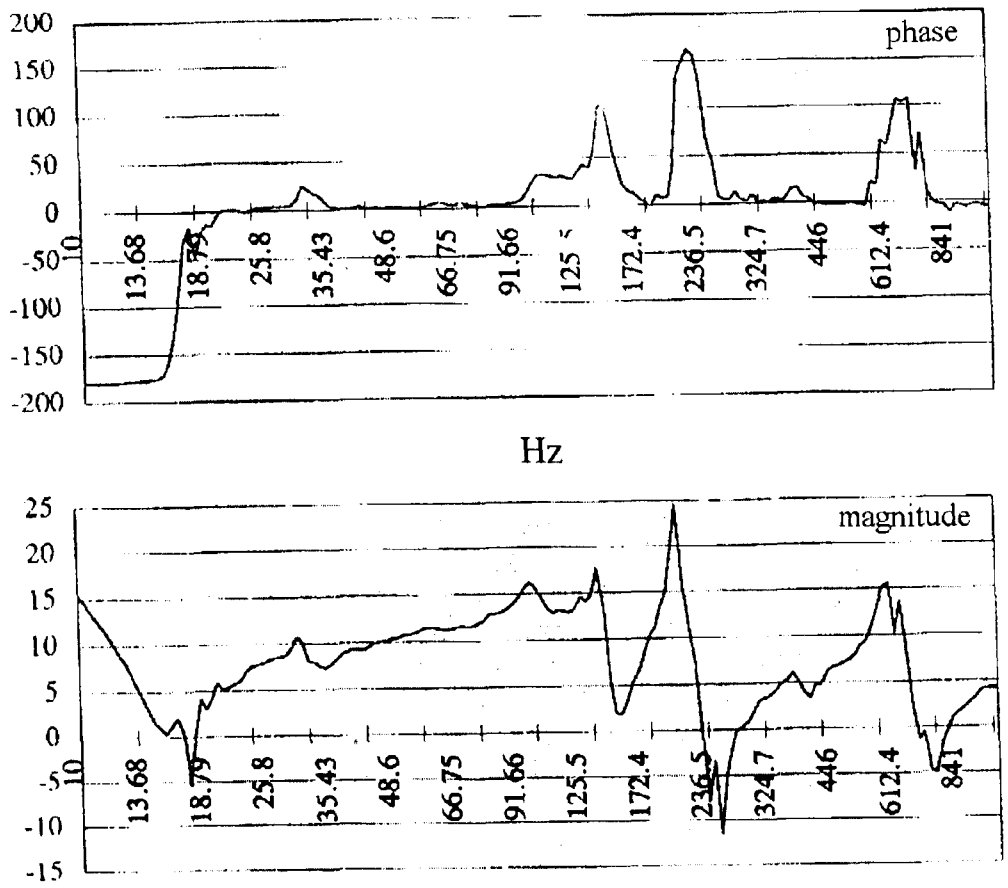

Figure 3.15: Measured frequency response data using the FRA2 method

at 15 Hz was applied at the sensor output (measurement).

Simulated frequency response - reflected tip position

Reversal of the polarity of the measured output transfer signal results in the response as depicted in Fig. 3.21. The resonance observed in the previous section is now an anti-resonance at the same frequency (14 Hz).

Tables 3.4, 3.5 and 3.6 give the complex eigenvalues (18) and zeros (16) in ascending order.

The tables above show that the lightly damped poles (baring one at -23.72) and zeros for both measurements are located close to the imaginary axis. This observation is consistent with that of other workers who have investigated non-collocated flexible link systems. It can be seen that both types of output measurement investigated have the same eigenvalues but the locations of the open loop zeros differ. The reflected output measurement is seen to have more RHS zeros than the arclength measurement. However, the reflected tip measurement is more stable at the lower frequencies of interest[6] since the RHS zeros for the reflected tip measurement are closer to the imaginary axis than those of the arclength of the tip measurement. Section 3.6.2 investigates the open-loop root locus of each measurement under simple PD control.

[6]in practical systems the higher modes are usually damped out due to the real effects of friction

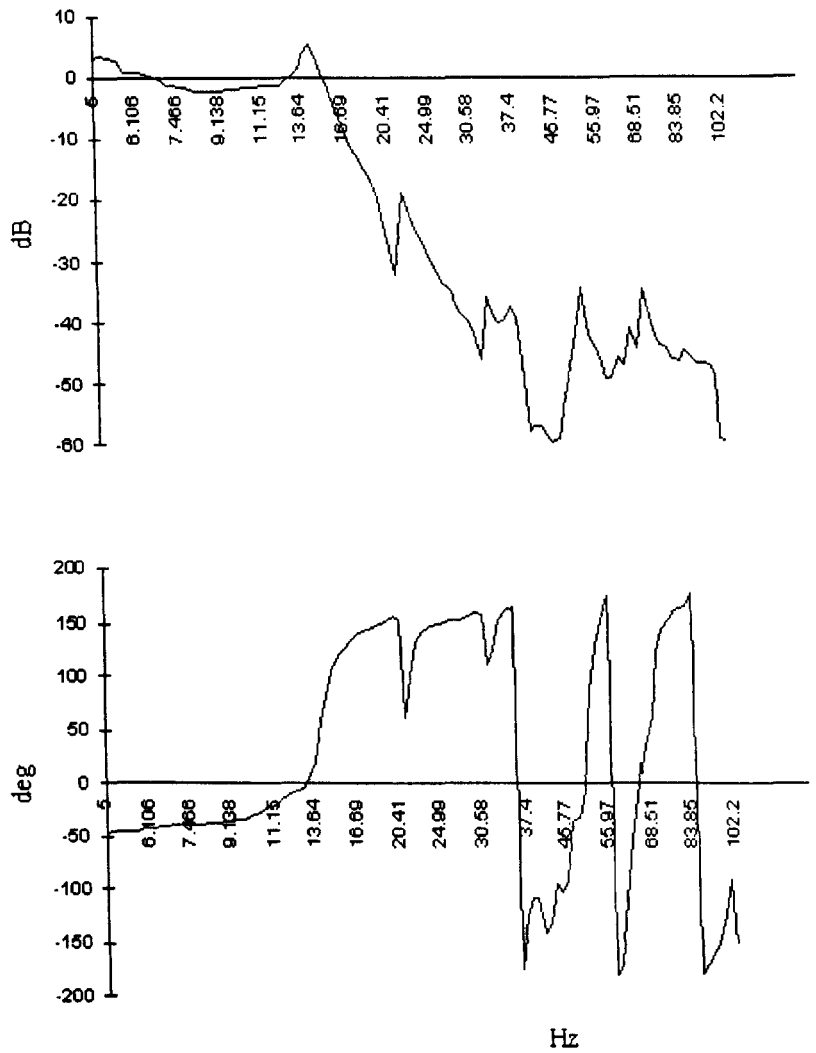

Figure 3.16: Measured frequency response data using the FRA1 method

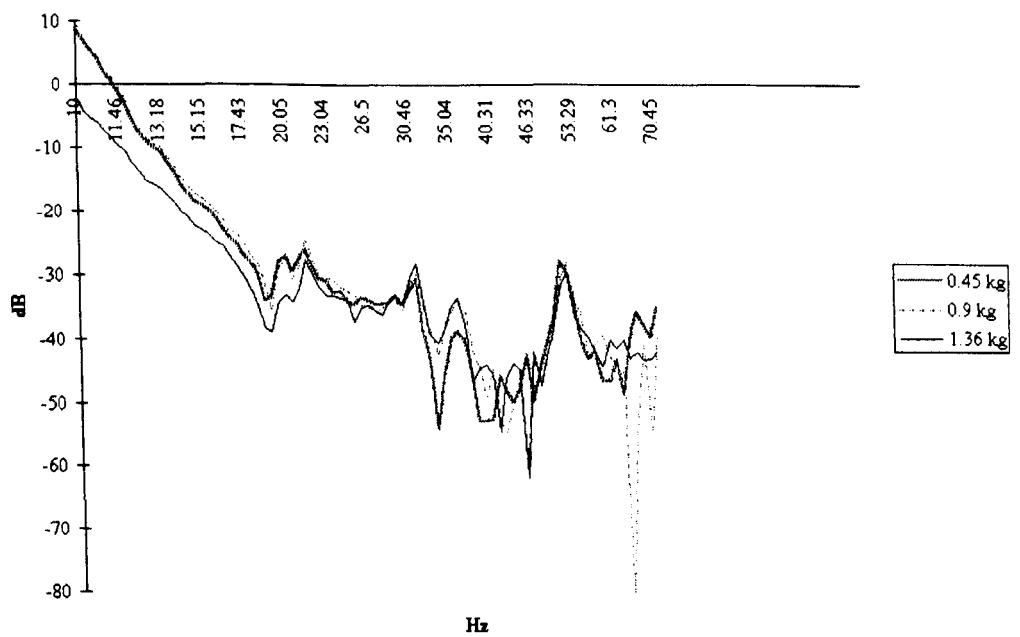

Figure 3.17: Loaded arm frequency response data using method FRA 1

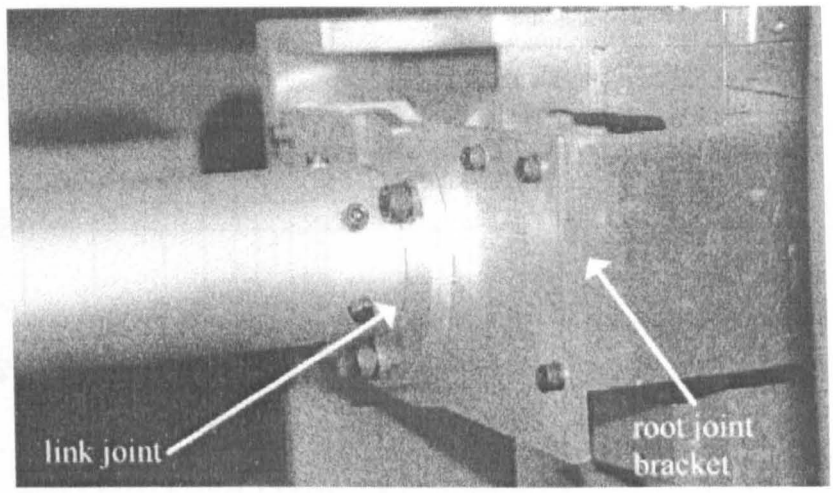

Figure 3.18: Link joint and bracket joint

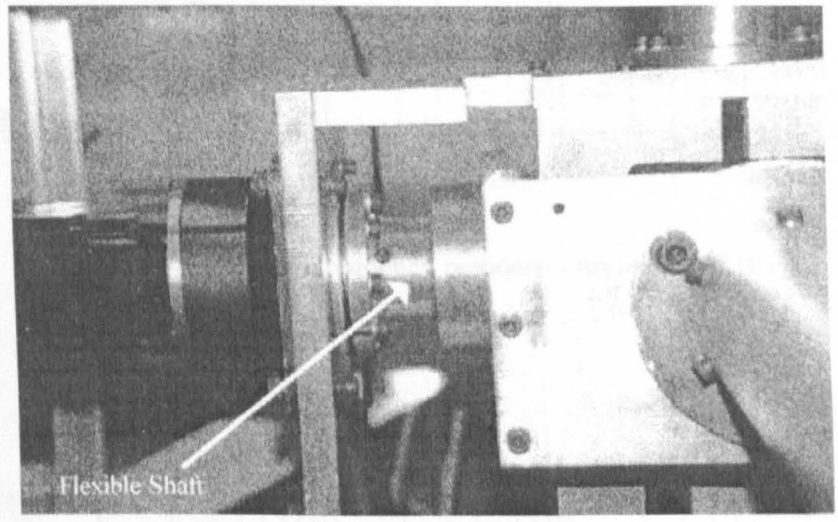

Figure 3.19: Flexible joint

Real	Imaginary	Frequency	Damping
3.4606E-09	±3204.99	3204.99	-1.08E-12
1.3724E-08	±1989.82	1989.82	-6.90E-12
0.			
0.			
0.			
0.			
0.			
0.			
0.			
0.			
0.			
-23.72			
-3.1951E-05	±100.03	100.03	3.19E-07
-7.1126E-08	±741.99	741.99	9.59E-11

Table 3.4: Eigenvalues for the arclength and reflected tip measurements

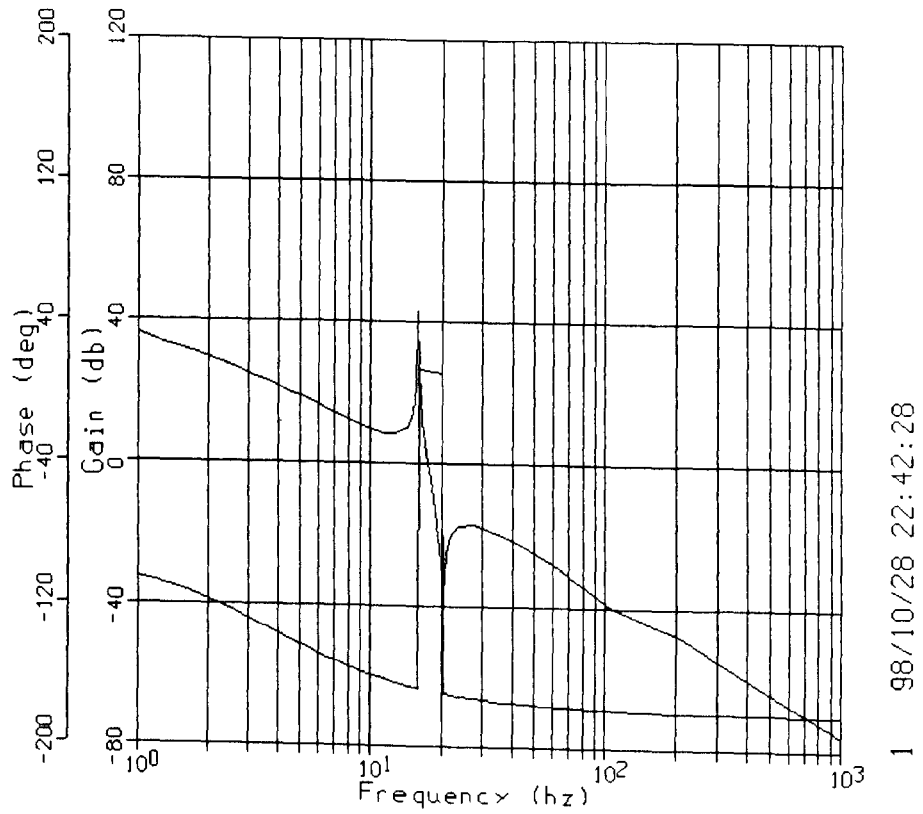

Figure 3.20: Simulated Frequency response - arclength of the tip

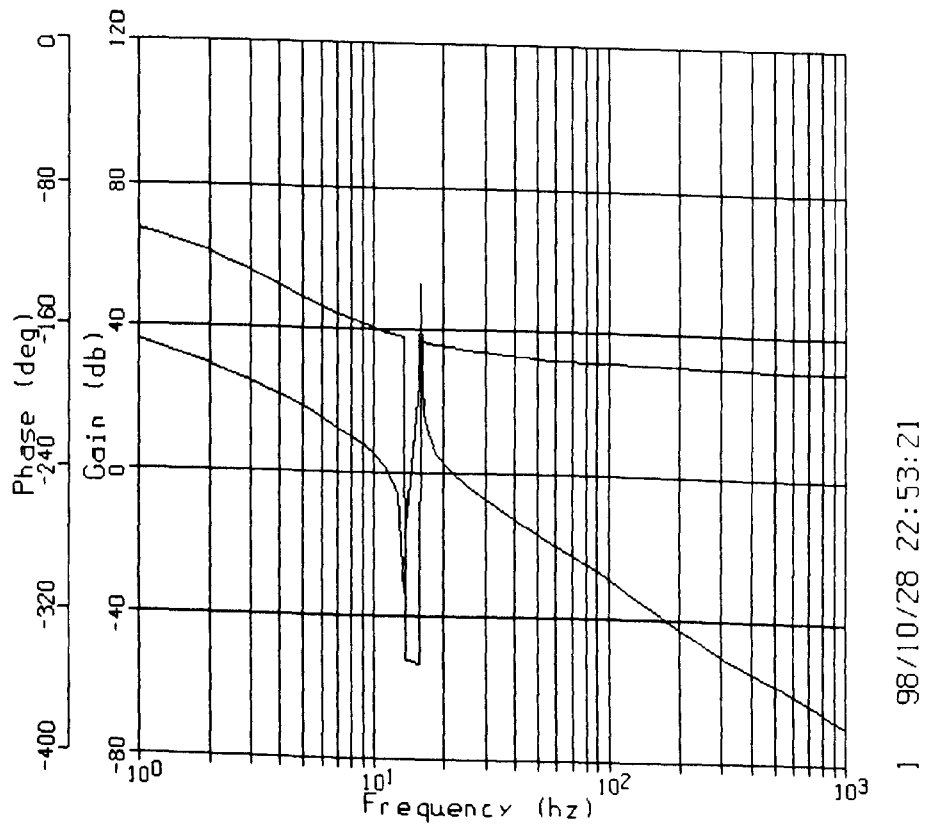

Figure 3.21: Simulated Frequency response - reflected tip position

Real	Imaginary	Frequency	Damping
3.1850E-14			
-3.9682E-10			
5.5616E-06			
-5.5616E-06			
3.7659E-10	±2.9059E-05	2.91E-05	-1.30E-05
-3.3834E-10	±3.5116E-05	3.51E-05	9.63E-06
-5.8762E-05	±127.699000	127.70	4.60E-07
1.0505E-06	±714.644000	714.64	-1.47E-09
-2.2646E-05	±2010.59000	2010.59	1.13E-08
1.1845E-05	±3197.25000	3197.25	-3.70E-09

Table 3.5: Zeros for the arclength of the tip measurement

Real	Imaginary	Frequency	Damping
0			
-2.9643E-14			
3.5971E-14			
-2.9859E-11			
1.4394E-08			
-1.4394E-08			
2.0409E-10	±2.00E-05	2.00E-05	-1.02E-05
2.5993E-05	±84.98	84.98	-3.06E-07
2.4074E-06	±754.99	754.99	-3.19E-09
9.9756E-06	±1979.69	1979.69	-5.04E-09
-5.3133E-06	±3208.66	3208.66	1.66E-09

Table 3.6: Zeros for the reflected tip measurement

3.5.2 Transient step response

A series of transient response tests about the vertical axis (no gravity) under classical P control were carried out to compare the simulation to the real robot. The observed responses are important for the examination of the performance criteria of the controlled system as well as providing data for the design of the fuzzy controller as described in subsequent chapters. Measurements of steps in both directions of the available degrees of freedom were made (LHS denoting movement leftward from the vertical axis motor looking out towards the arm and RHS denoting the opposite direction). Measurements were also taken for different masses placed at the tip of the arm, viz., 0.7, 1.15 and 1.6 kg. Figs. 3.22, 3.23 and 3.24 show the measured and simulated responses.

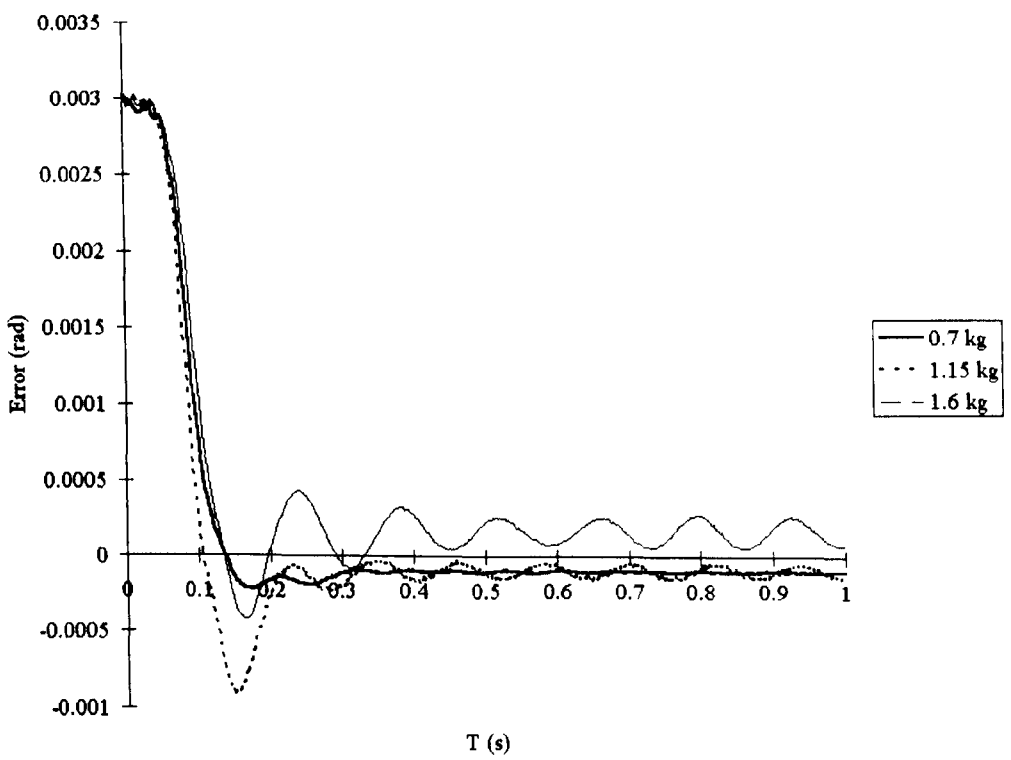

Figure 3.22: Measured transient responses with linear P control - LHS movement.

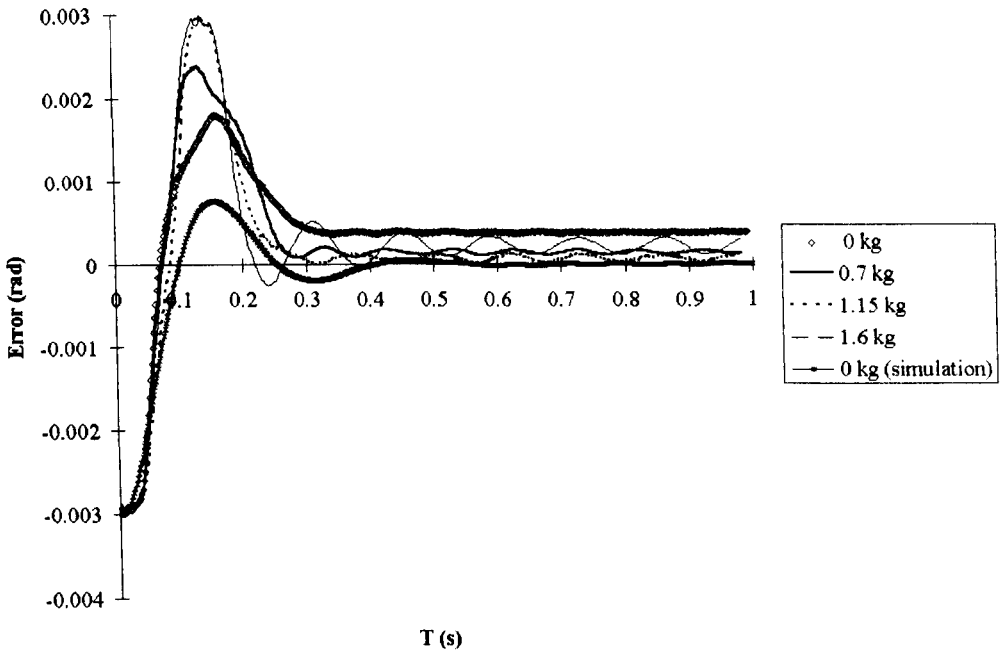

Figure 3.23: Measured transient responses with linear P control - RHS movement.

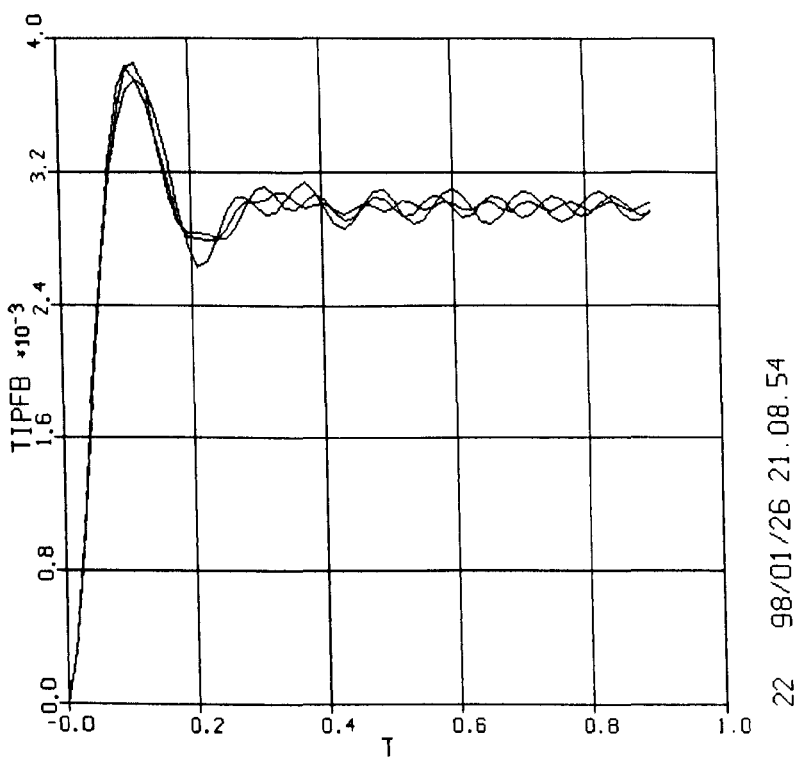

Figure 3.24: Simulated transient responses with linear P control.

Table 3.7 below summarises the observations for the different step responses. The mass of the sensor (= 0.154 kg) was included in all the measurements. The no payload case (0 kg) comprised of the mass of the flexible link and the mass of the sensor.

	M_{pt} %			T_s (s)			T_r (s)		
Mass (kg)	RHS	LHS	Simulation	RHS	LHS	Simulation	RHS	LHS	Simulation
0	60	–	–	0.34	–	–	0.05	–	–
0.7	80	6	28	0.4	0.32	0.4	0.05	0.05	0.065
1.15	100	13	27	0.4	0.5	0.5	0.055	0.055	0.065
1.6	96	30	23	> 0.8	> 0.8	> 0.8	0.05	0.055	0.065

Table 3.7: Observations on measured step responses under P control

The measurements above show a distinct difference between the transient response for the different directions of the applied step. Steps rightwards show a much higher initial overshoot than the leftward step responses. This difference can only be attributed to non-linear friction and stiction. There is evidently a greater component of coulomb friction in the leftwards direction which results in a more damped response even with a 1.6 kg payload. The rise-time and settling-time, T_r and T_s respectively, compare well in both instances. Each response tending to become more oscillatory as more payload is added.

The simulated response shows the characteristic initial high overshoot exhibited by the measured responses but the values of M_{pt} do not compare well. The RHS measurements are generally much higher than the simulation (a maximum deviation of 73 % occurs for 1.15 and 1.6 kg cases) and the LHS measurements are generally lower (maximum deviation of 22 % for the 0.7 kg case and a minimum deviation of 7 % for the 1.15 kg case). It is possible that these differences arise due to the different amounts of coulomb friction for steps in the different directions. The simulation only takes in account viscous rather than coulomb friction.

The effects of backlash are also prevalent in the experiments and large steady state errors are observed for the steps in either direction, with a maximum of 0.36 mm for a RHS step and no payload. The hysteresis effects due to backlash[7] could also be the reason for the inconsistency in overshoot behaviour for the different payload cases for steps in the LHS direction (the 1.6 kg payload case is observed to have a lower peak overshoot than the 1.15 kg case (c.f. table 3.7).

The simulated rise-time and settling-times compare favourably with the measured data for both step directions. Despite the differences in the measured and simulated overshoot due to unmodelled friction and backlash effects there is general agreement in the simulated and measured transient behaviour. Both simulated and measured responses show an initial high overshoot which is greatly damped by the time the second oscillation occurs. Also as the responses become more oscillatory as payload is increased.

3.6 Model analysis

In this section the model is analysed for robustness and stability. Simple classical P and PD controllers are investigated. Robustness of the controllers is determined through observing the system transient and frequency response behaviour when the system parameters are varied. Stability of the classical controllers is verified by observing the root-locus and transient and tip deflection response plots for the two controllers tested.

3.6.1 Parameter variation - Variation of step size

Lewis observed that the gradient or slope of the transfer function, a non-linear saturating function, varies with the size of the impinging light spot upon the quadrant detector [5]. He recommended a spot size between 50% and 75% of the detector size for robot operation as it gave the most linear

[7] due to the extra 1:10 gear ratio (toothed gears) that was added onto the 1:100 Harmonic Drive reducer c.f. section 3.4.1

response and highest voltages before saturation. Fig 3.25 depicts the sensor transfer function for spot sizes of half, three quarters and the full detector diameter.

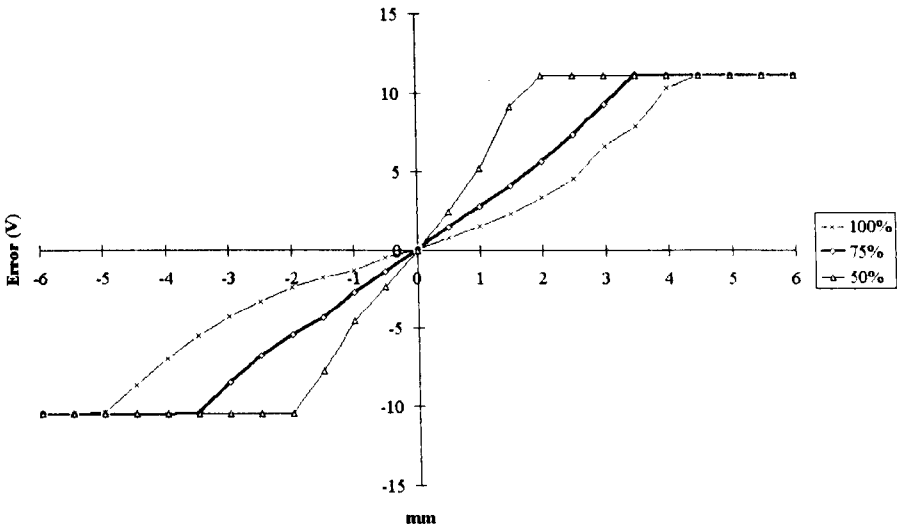

Figure 3.25: Measured optical sensor response for different sizes of impinging light spots

A spot size that is three quarters the diameter of the detector limits the maximum step size to 3mm before sensor saturation. Figs. 3.26 and 3.27 show the simulated results for step sizes of 0.03 and 1 radians respectively for an unloaded arm[8]. It can be seen that current saturation begins to set in even with the lower amplitude (0.03 radian) step. Fig. 3.28 depicts the results for the step size (constrained by sensor size) being used by the prototype.

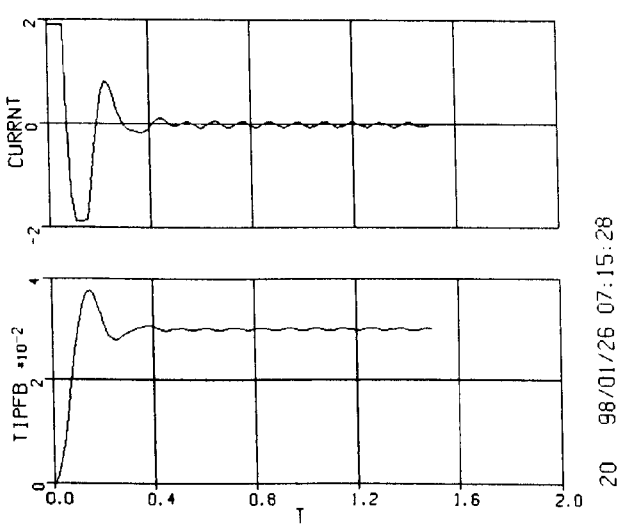

Figure 3.26: Input step size of 0.03 radians

[8]in these experiments the sensor saturation limits were removed

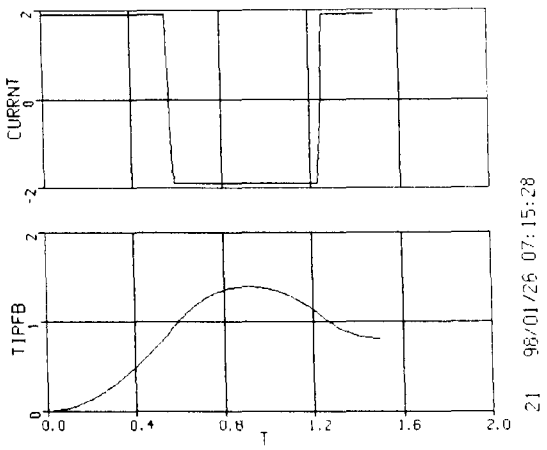

Figure 3.27: input step size of 1 radian

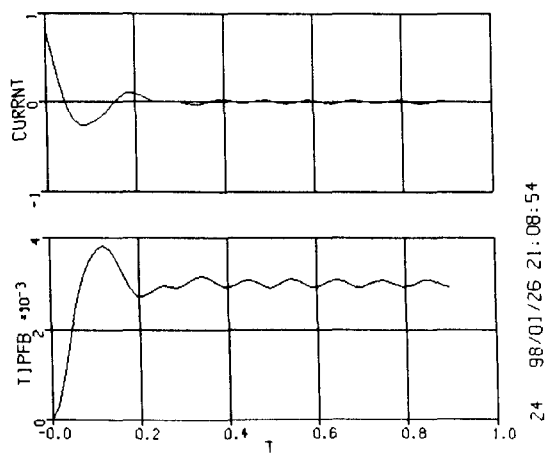

Figure 3.28: input step size of 0.003 radians

From the observations above it is quite clear that the variation of the input step size affects the system performance. It can be seen that the motor current saturates for 0.03 and 1 radian input steps. However the current remains well within limits when the step size is 0.003 radians. It can be seen that small step sizes are required in order to prevent current saturation. This observation will set a precedent for the choice of tip sensor and also justifies the present choice.

3.6.2 Stability of the model under simple classical control

This section will describe the tests carried out to verify the stability of the model and a simple PD classical controller. Models with *arclength* of the tip and *weighted*[9] *reflected output* measurements as described earlier, are investigated here. The stability of the model is determined by examining

[9]the weighting is normalised to 1 by using a 0.003 radian step input

the root locus and transient and deflection response of the of the controlled system. The root locus analysis is performed upon a piece-wise linearised transfer function obtained from ACSL.

Root locus and transient response analysis

Root locus analysis is a convenient method of examining the stability of a system under control as the gains of the controller are varied. The locus of the poles as the loop gain is varied gives an indication of the relative stability of the system. From this analysis it is possible to determine a 'safe maximum' value of loop gain before instability sets in. This maximum value is an important factor in the design of the stable supervisory fuzzy PD controller proposed in this thesis. Figs. 3.29 and 3.30 give the open loop root locus responses of the flexible link system using a PD controller with tip arclength and weighted reflected modified outputs respectively.

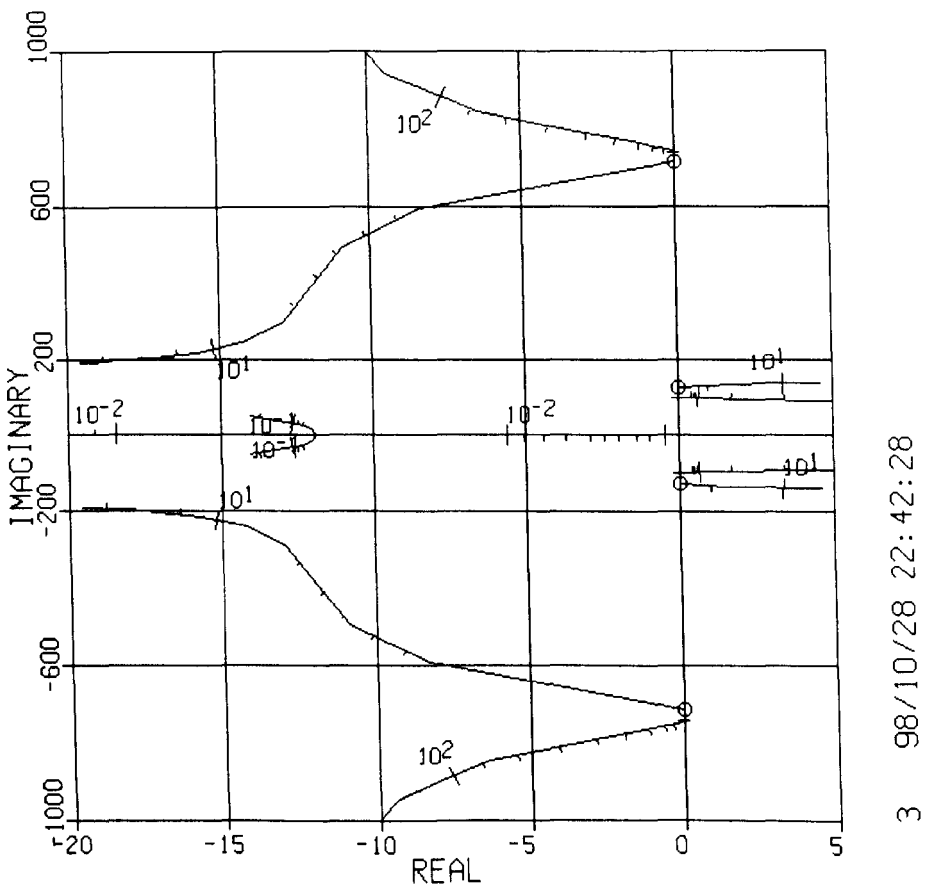

Figure 3.29: Root locus response for a PD controller - arclength output measurement

From the figures above it can be seen that both the arclength measurement and reflected tip measurement have zero dynamics in the RHS s-plane. However the margin of stability is much greater for the latter type of measurement. Higher gains (greater than 700 as opposed to 100) are required to excite zero dynamics for the reflected tip measurement.

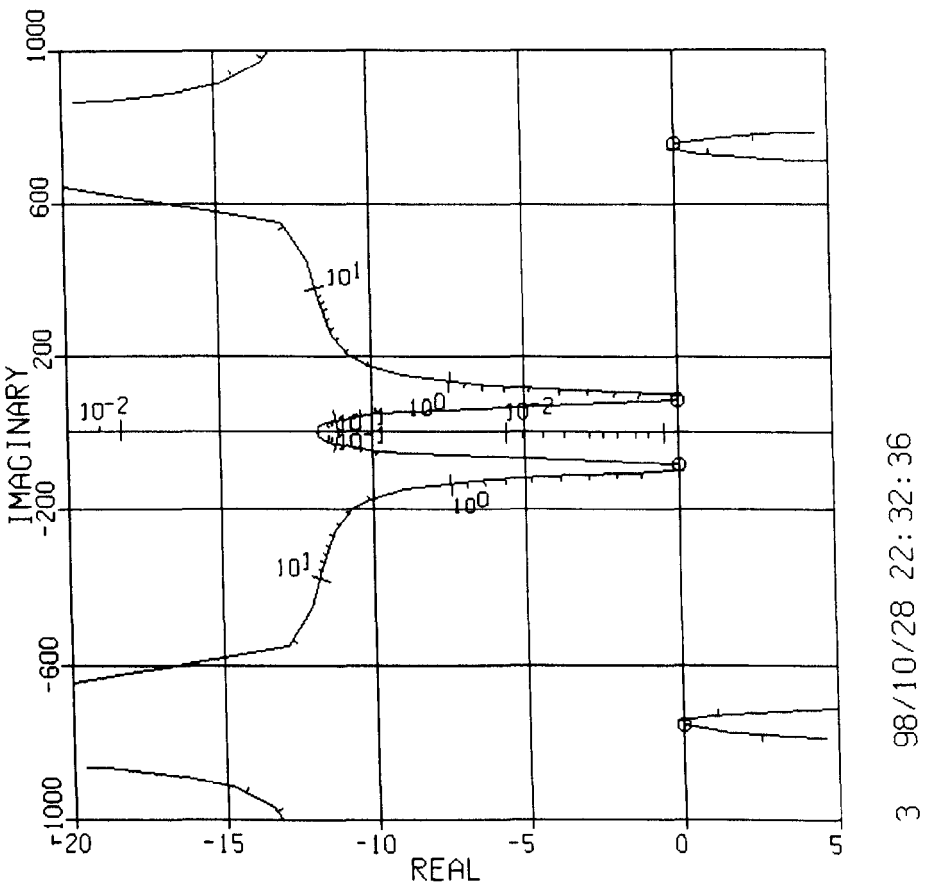

Figure 3.30: Root locus response for a PD controller - weighted reflected output measurement

3.7 Discussion and summary

This chapter has investigated the finite difference modelling of the TFS based robot. Parameters for the model, including the actuator and arm structure, have been determined form measurements carried out on the prototype robot.

The following observations have been made:-

- These measurements reveal that the range of the parameters can vary greatly depending on the configuration of the robot. The dynamic parameters i.e., moment of inertia of the arm and viscous friction vary by up to 37% and 43% respectively for a difference of 40° in arm elevation.
- Frequency analysis reveals that the source of the dominant low frequency vibrations can be attributed to mechanical flexibility at the link joint and bracket joint and flexure in the joint due to Harmonic Drive gearing. Adjustment of the model to incorporate the former source of flexure resulted in a simulated frequency response with a similar characteristic to the measurements. Both the simulation and measurement showed an underdamped fundamental mode at 14 Hz.
- In the transient response tests, the rise-time and settling-time performance of the simulation compared favourably but overshoot differed greatly. In practice the peak overshoot was shown to vary significantly with the direction of the applied step. These effects are attributed to coulomb friction and stiction. The simulation, which does not account for coulomb friction, was shown to have values of peak overshoot that were in between the values obtained for the different direction steps and different loads.
- The simulation was used to investigate both arclength of the tip and weighted reflected output measurements. The later measurement was show to be more stable than the arclength measurement. At the low frequencies of interest the transfer function obtained by using the reflected output measurement was shown to be minimum-phase. Root locus analysis showed that there was a finite gain before the transfer function of each of the output measurements displayed RHS zero dynamics. The reflected tip output measurement was shown to have a greater margin of stability than the arclength of the tip measurement. These observations are consistent with those of other workers who have studied different output measurements.
- Investigations into the robustness of the model under simple classical control have also shown that size of the sensor, or ultimately the size of the measured input error to the control system affects stability. Large step sizes have been shown to cause the actuator current to saturate. In contrast, the small 0.003 radian step applied to the prototype (constrained by sensor size and transfer characteristic), has been shown to be stable.

The exercise of developing a model of the TFS based system has resulted in a model that approximates the behaviour of the practical system. Although the transient response and frequency responses are in general agreement, even under different payload conditions, the model could not be used reliably for a model based controller. This is because there is still discrepancy in the transient and frequency response, due to un-modelled effects such as coulomb friction, flexure and backlash in the gearing and parameter variations due to the configuration of the robot. However, the approximate model has been vehicle to the detailed study of the system and can be used to test different control algorithms allowing predictions of their behaviour and insight into their stability.

The next chapter evaluates the different 'non-model' based control methods that are emerging and puts forward the fuzzy controller as one that best meets the specification laid down in Chapter 1.

References

[1] White A. S., Gleeson P. T., and Wong Y. *Simulation of Robot Vibrations*, pages 553–568. Clarendon Press, Oxford, 1993.

[2] Warburton G. B. *The dynamical behaviour of structures.* Pergamon Press Ltd., 1964.

[3] Editor. *Advanced Continuous Simulation Language ACSL Reference Manual.* MGA Software, USA, 1995.

[4] De Scutter J. and Van Brussel H. *Lecture notes of the short course on computer controlled motion - Third Edition.* Katholieke Universiteit Leuven, Faculty of Applied Sciences, Department of Mechanical Engineering, Machine Design and Automation (PMA), Celestijnenlaan 300 B - 3001 Heverlee - Belgium, 1992.

[5] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[6] Smith G.D. *Numerical solution of partial difference equations: Finite difference methods.* Clarendon Press, Oxford., 1979.

[7] Kuo B. C. *Automatic control systems - sixth edition.* Prentice Hall International, Inc., Englewood Cliffs, New Jersey 07632, 1991.

[8] Dorf R. C. *Modern Control Systems.* Addison-Wesley, USA, 1989.

[9] Millman J. and Grabel A. *Microelectronics.* McGraw-Hill, 1988.

[10] Klafter R. D., Chielewski T. A., and Nagin M., editors. *Robotic engineering : An integrated approach.* Prentice Hall, Inc., 1989.

[11] Chodavarapu P. A and Spong M. On noncollocated control of a single flexible link. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1101–1106, 1996.

[12] Wang D. Comparison of optimal and non-optimal control strategies for the single flexible link. *International journal of robotics and automation*, 9(3):130–136, 1994.

[13] Pota H. R. and Vidyasagar M. Passivity of flexible beam transfer functions with modified outputs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2826–2831, 1991.

[14] Wang D. and Vidyasagar M. Transfer functions for a single flexible link. *IEEE Trans. on Robotics and Automation*, 5(3):373–377, 1989.

[15] Tarn T. J., Guo C., and Bejcy A.K. Issues in zeros of flexible robot systems. In *Proceedings of the* 32^{nd} *Annual Allerton Conference on Communication, Control and Computing*, pages 641–650, 1994.

[16] Pape M. An investigation of the structural and dynamic properties of an optically guided arm, Erasmus project, Middlesex University, London, England. 1995.

[17] Swevers J., Torfs F., Demeester F., and van Brussel H. Fast and accurate tracking control of a flexible one-link robot based on real-time link deflection measurements. *Mechatronics*, 2(1):29–41, 1992.

Chapter 4

Application of fuzzy control to the TFS based robot

4.1 Introduction

This chapter gives a brief overview of Artificial Intelligence methods from a control viewpoint. AI techniques are found especially useful in control applications where the process cannot easily be modelled mathematically and where its behaviour is essentially non-linear and unpredictable [1]. The previous chapter showed that such is the behaviour of the TFS based robot being investigated in this thesis.

Both fuzzy and artificial neural network (ANN) systems can be used for the real-time control of non-linear processes and are able to satisfy the requirements defined by the specification for the TFS based controller set in Chapter 1. The development of the former approach is investigated in this thesis. The choice of this method stems from the requirement of a safe and robust controller in a safety critical environment. However, the ANN approach is not completely dismissed and Chapter 8 describes some initial investigations.

The aim of this chapter is to give a background into fuzzy set theory required for design of the fuzzy based controller and to establish the structure of the fuzzy controller. The choice of the controller structure is based on application in a safety critical environment, the experience of previous researchers on controlling the TFS based robot and the requirement of implementing the controller in an embedded real-time environment.

4.1.1 Intelligent control

Traditionally control algorithms have been devised based on the formalism of mathematical descriptions of the process behaviour [2]. In optimal control, a cost function or performance index, which is derived from the linearised mathematical model, must be minimised to achieve desired performance. A closely coupled design procedure between the control algorithm and analytic solution of the model is implied.

'Loosely coupled' design procedures are used when a linear model is not available or the control performance is not easily translated into simple criteria and cost functions. In such cases, where no analytical solution exists, the design is translated to a numerical optimisation problem and there is more freedom in the design process. Empirical data, obtained through simulation or experiments on the process, is often used to support the choice of controller structure and parameters. Here, mathematical models are still used, but the knowledge and experience of the designer are necessary in the final design stage.

In contrast to the methods above, implementation of the controller in TFS based control is completely uncoupled from a mathematical description of the plant and no constraints are placed on the choice of the type of model or performance criteria. The tip positioning error to be minimised is obtained through the physical configuration of the system as described in Chapter 1. In its most

basic form TFS based control reduces to a simple linear feedback regulator. In such a case, where solution of a mathematical model is not required to achieve the desired end effector positioning, methods based on the Artificial Intelligence AI [3] approach may be used to improve control performance. The aim of using AI techniques is to devise a controller based on observations of the plant behaviour i.e., if the process is described by a qualitative model or if the controlled process scenario is variable and not well defined [4].

Systems which adaptively estimate continuous functions from data without specifying mathematically how outputs depend on inputs can be described as 'intelligent' model free estimators [5]. Several AI based control methods have emerged in situations where methods based on conventional control theory cannot be used even when enhanced by either adaptation of the controller or when the design is based on robust control methods [2]. A taxonomy of such systems is given in Fig. 4.1.

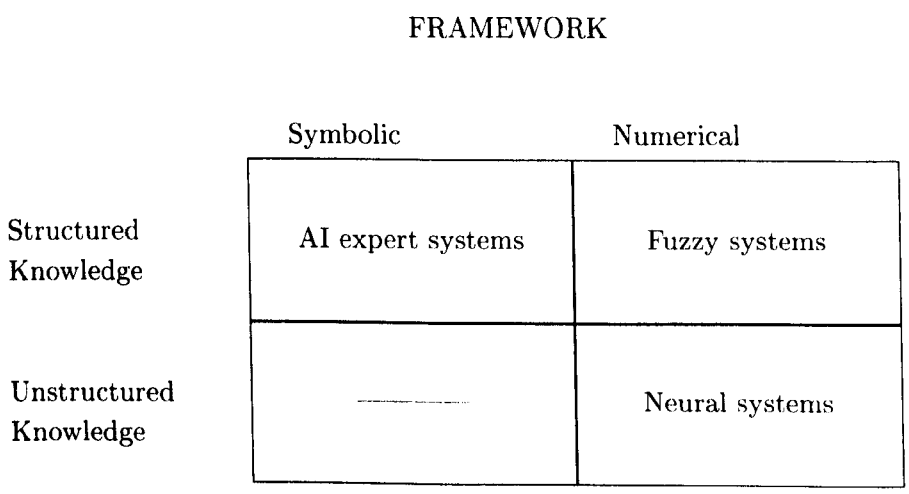

Figure 4.1: Taxonomy of model-free estimators [6]

The main AI based methods are characterised by [4]:-

Symbolic and problem solving approach Knowledge based systems based on expert systems - These may be described as collections of 'rules of thumb'. Processing of the rules is undertaken by an inference engine that enables rules to be triggered based on patterns defined by occurrence of previous conditions rather than calls to functions. The framework for processing these rules is symbolic rather than numeric and most classical procedural based programming languages are unsuitable. Specialised languages, such as PROLOG, LISP and SMALLTALK-80 [7], are used. The 'structured' or rule-like methods of reasoning of expert systems are attractive to control applications because they allow the control actions to be performed as a result of rule evaluation or reasoning. A human operator's experience in controlling a process is easily expressed within this framework. Expert systems have been successfully applied to fault detection [8], fault diagnosis [9], supervisory control [10] and in process identification for the design of controllers. However, because processing is carried out within a symbolic framework, there is difficulty integrating the expert system into the process control environment as numerical algorithms, sensor based acquisition and real time issues need to be dealt with. Also expert systems make little provision for the handling of uncertainty since the rules purely empirical and are specified by the designer.

Learning capability Artificial neural networks attempt to model the function of biological neural systems [11]. The intelligence of these systems arises from the ability of the network to learn, generalise and classify. A wide range of ANN models and methods of training exist. Lippmann [12] and Khanna [13] have reviewed the many different topologies and Kung [14] gives a detailed description of different training and optimisation methods applied to ANNs.

The learning capability of ANNs to learn and generalise and perform non-linear mappings makes them suitable for the control of non-linear systems which cannot be easily modelled mathematically [15]. They are also attractive for control applications because they operate within a numerical framework and their parallel processing nature which gives them a high fault tolerance and high processing throughput which is useful in real-time applications [16]. Concerns, however, exist about their robustness.

The 'intelligence' of an ANN system is stored in a set of weights which may be determined through training either off-line or on-line. The output of the ANN controller is usually a summation of these weights passed through a non-linear activation function which models biological synaptic 'firing' [17]. This 'black-box' nature of a ANN presents a drawback in terms of stability and performance evaluation of the system as the behaviour of the individual weights and how they affect the system outputs is not easily discerned. Also, there is the dynamic nature of the weights to contend with. Pegman [18] of the National Advanced Robotics Research Centre (NARRC) states that *a strong design methodology that supports the designers reasoning and facilitates easy third party inspection and validation is required in the design of safe robots.* ANNs are not dismissed entirely with regard to safety. In the case where a detailed hazard analysis and risk analysis of the system cannot be adequately carried out to partition the safety critical functions of the system, use of a 'harness' to monitor parameters associated with direct hazards can be used. The study above will form the basis of future research.

Models with implicit approximated knowledge Fuzzy systems based on fuzzy sets and logic - These systems differ from expert systems in that symbolic knowledge representation is processed within a numeric framework. The existence of this compiled level is the basis for fast real-time control implementations [19]. Fuzzy systems are based on the concept of multi-valued or 'grey' logic as opposed to a binary 'on-off' logic. The fuzzy set is a manifestation of multi-valued logic when applied to a linguistic variable. Linguistic variables such as 'tall', 'hot', 'far', 'fast' etc. can be modelled as fuzzy sets. These sets can be manipulated by using fuzzy set theory, developed by Zadeh [20][21][22], to result in a decision or inference based on a store of knowledge (the fuzzy relation between fuzzy sets) that is inexact. Thus, using fuzzy logic it is possible to express a human operators experience, which has been articulated in natural language, within a numeric framework.

Another difference between Expert systems and fuzzy systems is that the latter are able to deal with uncertainty [23][24]. Expert systems require detailed rule elicitation, i.e., the conditions being modelled must be exact. Effective modelling of a process to be controlled using expert systems requires a large number of rules, especially if the conditions are dynamically changing, this further compromises processing speed. It is also found that the knowledge base operates more smoothly when the rules are based on fuzzy sets and logic [2] [25].

4.2 Fuzzy set theory

Human operators often succeed in control situations where an automatic controller cannot be easily implemented. They are able to control highly non-linear systems using imprecise information based on observation of complex patterns of measurements and unmeasurable variables such as colour, consistency etc. The design of a traditional controller requires the system to be controlled to be described in a rigid mathematical form and it is difficult to encode this imprecise information based on the human operators observations within a classical or modern control framework.

Zadeh [22] summarised the problems that occur in the control a highly non-linear system in the Principle of Incompatibility:

> *'As the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.'*

This principle can be used to explain why it would be difficult to use a traditionally designed

control system which defines its control outputs based on the precisely modelled characteristics of the system to be controlled. Fuzzy set theory can be used to encode the imprecise information that the human operator relies on within a numerical framework, in this way it is possible to develop a quantitative control system based on a human operators experience. The basis of set theory is the fuzzy set and the operations that can be carried out on them.

4.2.1 Fuzzy sets

Classical set theory relies on a binary 'on-off' or '1-0' relationship for its member variables, elements may either belong to the set or not. A crisp or non-fuzzy set defines this relationship depicted in Fig. 4.2a for the linguistic variable 'tall'. Using the fuzzy representation depicted in Fig. 4.2b it is possible to describe someone as tall to a certain degree rather than only tall or not tall. This representation is more realistic and enables us to account for uncertainty as the definition of tall is entirely subjective. In fuzzy set theory a fuzzy set, $A = TALL$, within a universe U, for example U = height, is characterised by a membership function $\mu_A(u)$, which assigns each element $u \in U$ a number $\mu_A(u)$ in the interval 0 to 1, that represents the grade of membership in A. A is completely determined by the set of tuples:-

$$A = (u, \mu_A(u)) \mid u \in U \tag{4.1}$$

In equation 4.1 | is interpretted as 'having the property' i.e., u and $\mu_A(u)$ have the property of belonging to U.

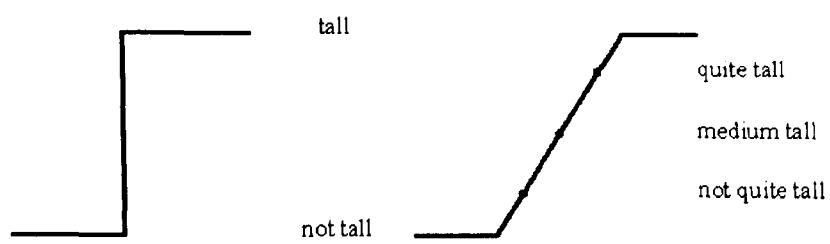

Figure 4.2: The binary and fuzzy representation of $A = TALL$

When U is finite or discrete universe of discourse A may be represented by:-

$$A = \mu_A(u_1)/u_1 + ... + \mu_A(u_n)/u_n = \sum_{i=1}^{n} \mu_A(u_i)/u_i \tag{4.2}$$

here '+' denotes an enumeration and satisfies $a/u + b/u = max(a,b)/u$. When U is uncountable or continuous the fuzzy set may be written as:-

$$A = \int_U \mu_A(u)/u \tag{4.3}$$

The shape of a fuzzy set may vary. Fuzzy sets with straight edges are commonly applied in control applications and may be easily described using the LR-notation from Dubois and Prade [26]. Some of the more common shaped fuzzy sets are described by functions the below:-

Definition 1.0 The function $\Gamma : U \rightarrow [0,1]$ is a function with two parameters defined

$$\Gamma(u; \alpha, \beta) = \begin{cases} 0 & u < a, \\ (u - \alpha)/(\beta - \alpha) & \alpha \leq u \leq \beta, \\ 1 & u > \beta. \end{cases} \tag{4.4}$$

Definition 1.1 The function $L : U \rightarrow [0, 1]$ is a function with two parameters defined as

$$L(u; \alpha, \beta) = \begin{cases} 1 & u < a, \\ (\beta - u)/(\beta - \alpha) & \alpha \leq u \leq \beta, \\ 0 & u > \beta. \end{cases} \tag{4.5}$$

Definition 1.2 The function $\Lambda : U \rightarrow [0, 1]$ is a function with three parameters defined as

$$\Lambda(u; \alpha, \beta, \gamma) = \begin{cases} 0 & u < a, \\ (u - \alpha)/(\beta - \alpha) & \alpha \leq u \leq \beta, \\ (\beta - u)/(\beta - \alpha) & \beta \leq u \leq \gamma, \\ 0 & u > \gamma. \end{cases} \tag{4.6}$$

Definition 1.3 The function $\Pi : U \rightarrow [0, 1]$ is a function with four parameters defined as

$$\Pi(u; \alpha, \beta, \gamma, \delta) = \begin{cases} 0 & u < a, \\ (u - \alpha)/(\beta - \alpha) & \alpha \leq u < \beta, \\ 1 & \beta \leq u \leq \gamma, \\ (\delta - u)/(\delta - \gamma) & \gamma < u \leq \delta, \\ 0 & u > \gamma. \end{cases} \tag{4.7}$$

Figure 4.3 describes the set above graphically.

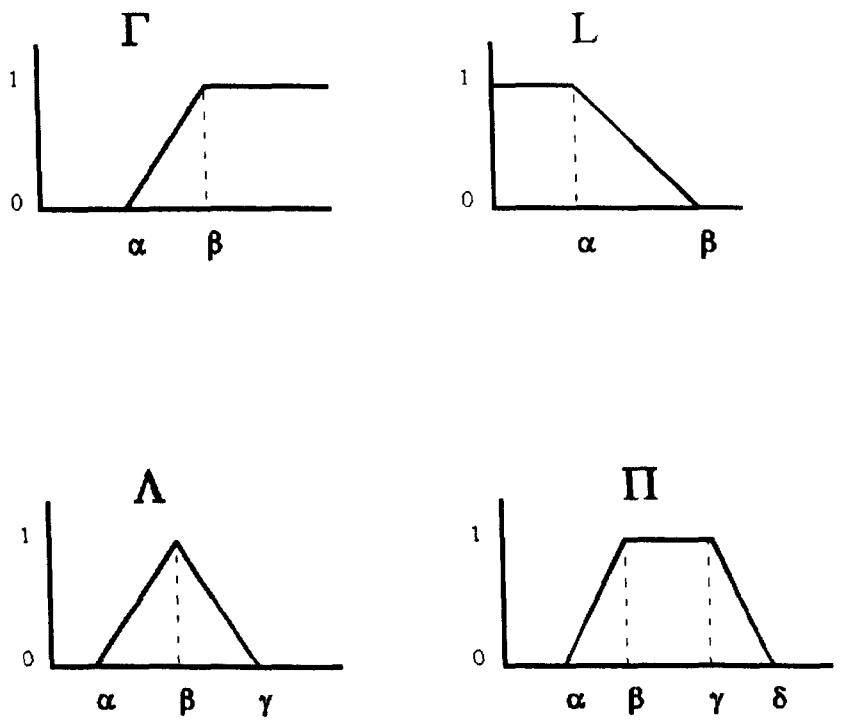

Figure 4.3: Shapes of the common fuzzy sets used in control

4.2.2 Operations on fuzzy sets

Fuzzy theory differs from classical set theory in that 'graded' sets as opposed to 'crisp' sets are used. The operators that act on these graded sets must be re-defined in order to account for the grey or fuzzy boundary of the sets. Zadeh published the first definitions of such operators in 1965 [27].

Intersection $\bigcap$

$$\forall x \in X : \mu_A(x) = min(\mu_A(x), \mu_B(x)) \tag{4.8}$$

An example of Eq. 4.8 is if $\mu_A(x) = 0.8$ and $\mu_B(x)$ 0.3, then $min(0.8,0.3) = 0.3$.

Union $\bigcup$

$$\forall x \in X : \mu_A(x) = max(\mu_A(x), \mu_B(x)) \tag{4.9}$$

An example of Eq. 4.8 is if $\mu_A(x) = 0.8$ and $\mu_B(x)$ 0.3, then $max(0.8,0.3) = 0.8$.

Complement $\sim$

$$\forall x \in X : \mu_{A'}(x) = 1 - \mu_A(x) \tag{4.10}$$

An example of Eq. 4.8 is if $\mu_A(x) = 0.8$ then $\sim(0.8) = 0.2$.

The more popular operations on fuzzy sets and those which are commonly used in control applications i.e., *union*, *intersection* and *complement* can be represented by a general set of functions known as generalised triangular norms [28].

4.2.3 Fuzzy Relations

Fuzzy relations are important in fuzzy logic to describe interaction between variables. If U and V are continuous universes of discourses, and $\mu_R : U \times V \rightarrow [0,1]$, then the binary fuzzy relation on $U \times V$ is defined as:-

$$R = \int_{U \times V} \mu_R(u,v)/(u,v) \tag{4.11}$$

For discontinuous universes the following definition is used:-

$$R = \sum_{U \times V} \mu_R(u,v)/(u,v) \tag{4.12}$$

The relation *much hotter than* can be represented as $\mu_R = \Gamma(x-y;0,30)$. Where *much hotter than* means a temperature difference greater than 30° Celsius.

$$\Gamma(x-y;0,30) = \begin{cases} 0 & for \;\; x-y < 0, \\ \frac{x-y}{30} & for \;\; 0 < x-y < 30, \\ 1 & for \;\; x-y \geq 30. \end{cases} \tag{4.13}$$

For X $\equiv$ Y = {10,20,30,40} this binary relation, defined in X $\times$ Y, can be represented in matrix form as:-

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.33 & 0 & 0 & 0 \\ 0.66 & 0.33 & 0 & 0 \\ 1 & 0.66 & 0.33 & 0 \end{bmatrix} \tag{4.14}$$

The triangular norms defined above can be used to combine different relations to give different meanings to the propositions. For instance, the fuzzy rule:-

if e is PB and $\dot{e}$ *is* PS, *then* $\dot{u}$ *is* NM

can be represented by the ternary relation, R:-

$$R = \int_{E \times \Delta E \times \Delta U} min(\mu_{PB}(e), \mu_{PS}(\dot{e}), \mu_{NM}(\dot{u}))/(e, \dot{e}, \dot{u}) \tag{4.15}$$

Here PB (positive big), PS (positive small), and NM (negative medium) are fuzzy sets defined on E, ΔE and ΔU respectively. In order to realise the above rule, both the fuzzy sets e *is* PB and $\dot{e}$ *is* PS are first combined by an AND operation or *conjunction* (can be done by using a *min* t-norm) and then composition is used perform the THEN part. This latter part is often described as rule firing. Composition is described in further detail in the following section.

4.2.4 Approximate reasoning and inference rules

A fuzzy expert system is an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert system are usually of a form similar to the one described in the previous section. Here the combination of the fuzzy if-then production rule through a relation defined on E, ΔE and ΔU may be seen as the it 'meaning' of the proposition or its *inference*. The antecedant (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables.

Under *inference*, the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Combinations of the triangular norms may be used as inference rules.

As with fuzzy operators there are many different inference rules that may be applied to derive the meaning of the production if-then rule described earlier. Of these, *the compositional rule of inference* [22][29] and the *generalised modus ponens* are perhaps the most important. The former uses fuzzy relations to define the explicitly connection between two propositions and the latter uses an if-then rule to imply a relation. The compositional rule of inference can be considered a special case of the generalised modus ponens. The main difference between them is that the compositional rule of inference always requires an explicit relation given apriori. The meaning of *if X is A then Y is B* is formally represented by the composition:-

$$B = A \circ R = proj(ce(A) \cap R) \text{ on } Y \tag{4.16}$$

where A is a fuzzy set defined on X and R a fuzzy relation defined on $X \times Y$. In eq. 4.16, the *proj* (projection) [30][31][32] and *ce* (cylindrical extension) are used to combine the one-dimensional vector representation of a discrete fuzzy set with the two-dimensional relation matrix. The meaning or *implication* of this composition depends on the type of relations used (as derived from, for example, the use of different t-norm operators).

With respect to fuzzy control, the most important type of implication is *Mamdani implication* [19][33]. In this implication relations are defined by using *min* for intersection and *max* for the projection that results in the output set. The relation *'if X is A then Y is B'* is defined as:-

$$\mu_{R_m}(x,y) = max(min(\mu_A(x), \mu_B(y))) \tag{4.17}$$

Under Zadeh implication the above relation is defined as:-

$$\mu_{R_m}(x,y) = max(min(\mu_A(x), \mu_B(y)), 1 - \mu A(x)) \tag{4.18}$$

Equations 4.17 and 4.18 describe the firing of a single rule. A set of rules can be combined in the relation matrix before composition so that for k rules:-

$$R_m = \bigcup_{k=1}^{n} R_m^{(k)} \tag{4.19}$$

Composition now results in an output set that represents the entire set of rules being fired. This method of inference is described in the literature as *Composition based inference.*

4.2.5 Individual rule firing

The previous section described how composition based inferences are derived from relations. This approach to inference is based on matrix manipulation as the rules or relations are encoded within a relation matrix. Matrix manipulation is costly in terms of processing especially in real-time control applications [34][35][36].

Fortunately other method of inference or 'rule-firing' exist which are less computationally expensive. These methods are been described by [19] as *Individual rule based inference* and are especially suited to 'look-up' table approaches that can be used in real-time fuzzy control.

Individual rule based inference can be achieved using a modified Mamdani implication:-

$$\forall u \in U : \mu_{B'}(u) = min(\mu_A(e^*), \mu_B(u)) \tag{4.20}$$

Here e^* is the fuzzified input and is obtained by:-

$$\forall e : \mu^*(e) = \begin{cases} 1 & for \ \ e = e^*, \\ 0 & otherwise. \end{cases} \tag{4.21}$$

Under Mamdani implication the 'firing' of a single rule would be depicted as in fig. 4.4.

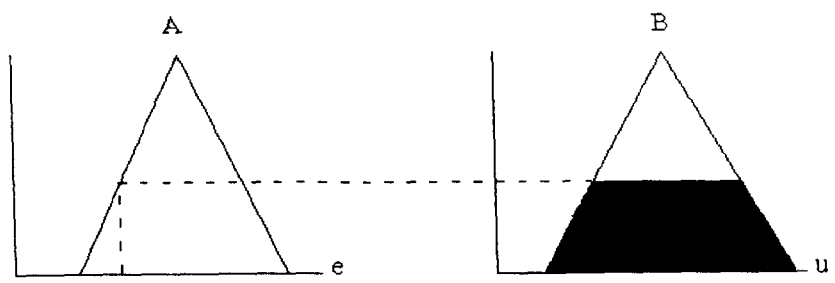

Figure 4.4: Individual rule firing - Mamdani implication.

Individual rule based inference can be carried out by another method as described below.

$$\forall u \in U \ : \ \mu_{B'}(u) = \mu_A(e^*) \cdot \mu B(u) \tag{4.22}$$

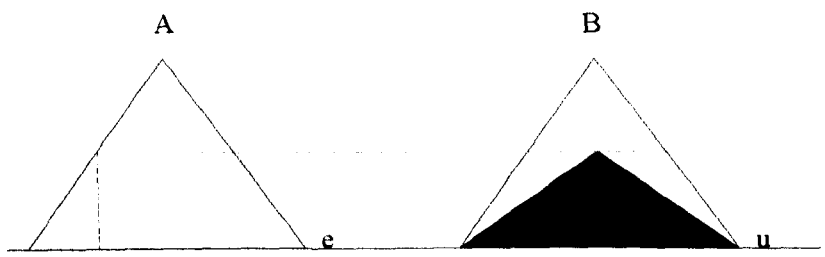

Figure 4.5: Individual rule firing - Scaled implication.

Eq. 4.22 results in a 'scaled' output set as depicted in fig. 4.5. The scaled sets for all the rules can then be combined into one aggregate fuzzy set by using the *max* t-norm. For discrete fuzzy sets this is done by taking the sum of the grades of membership in the B' sets to give an overall output fuzzy set which can be *defuzzified* to give a crisp output value. Defuzzification is described in a later section.

Kosko [37] refers to these inference methods as *correlation minimum* encoding and *correlation product* encoding respectively. The latter method preserves more information about the system inference since the data is not clipped [38].

The individual rule-firing method of inference using the *scaled* implication is particularly suited to real-time applications as processing-expensive matrices are not used in the computation of the inference. Also, the *product* operator results in a smoother output surface [39] and is particularly suited to systems which have output singletons as opposed to fuzzy sets.

The literature often refers to a third type of inference proposed by Takagi and Sugeno [40]. This method differs from those described above in that the consequent part is a function comprised of a linear combination of the input variables plus a constant rather than a set. Rule-firing in a manner similar to correlation product encoding would result in a scaled output of the consequent function, e.g. $\mu.f(.)$, instead of yielding a scaled output set.

4.2.6 Defuzzification

This is the final stage after composition in the inference process. Here the qualified consequents are aggregated to produce a crisp output. Different methods of defuzzification exist depending on

the type of inference and their implementation. Jang [41] classifies fuzzy reasoning into three types as depicted below in Fig 4.6.

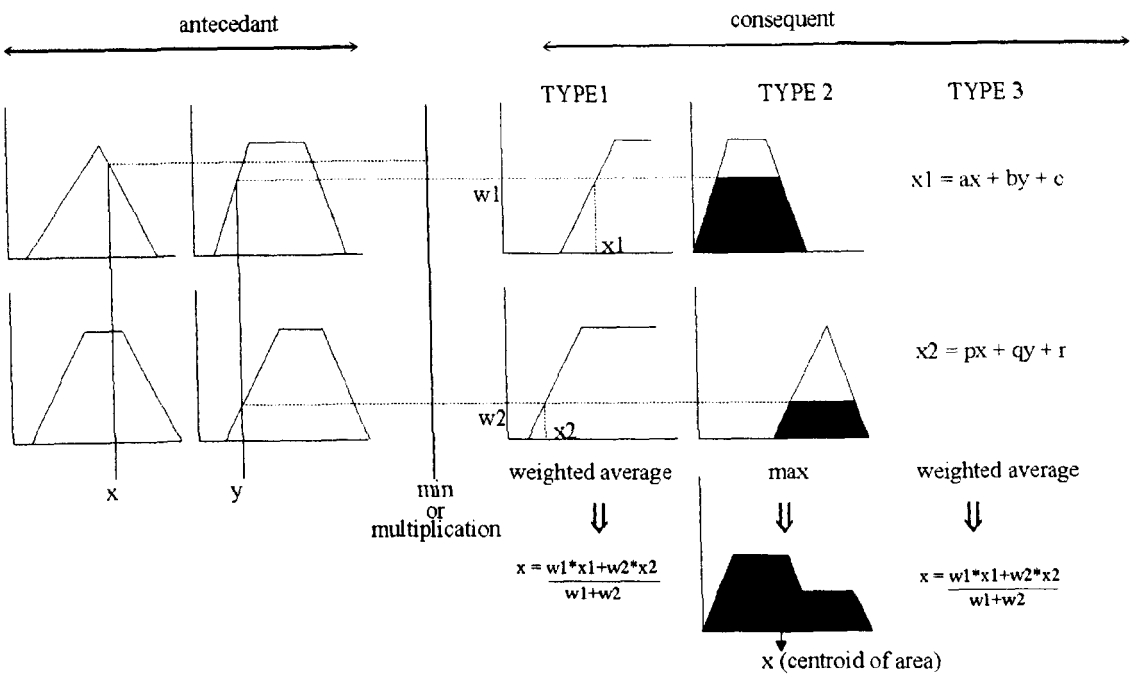

Figure 4.6: Commonly used fuzzy reasoning mechanisms [41]

The inference mechanisms differ in the consequent part. The output sets of a *Type1* inference comprise of monotonic functions or singletons [42], a crisp output is derived from a weighted average of the weighted singleton rule outputs.

For a *Type2* inference the output comprises of a continuous set and defuzzification is computation of a crisp output based on the summed area under the 'fired' output sets. Various schemes have been proposed to choose the final crisp output based on the overall fuzzy output. Examples are the centroid or centre of area, bisector of area, mean of maxima and maximum criterion [43][44][19][6].

Type3 inference is base on the Takagi-Sugeno method described in the previous section. Defuzzification is a weighted average of each rule's output.

The choice of the defuzzification method is important as it affects the quality of the control output and computational complexity. For *Type2* inference systems it can be seen that there is a wide choice of defuzzification methods available and the performance of each would have to be evaluated to determine the one best suited to the particular control task. The centre of gravity method or centroid is favoured by Brown *et al.* [39] for the following reasons:-

- It can be linked directly with artificial neural networks to make them adaptive (in the case where each individual rule can be weighted by a scalar w_j (where j is the j^{th} rule) which can be changed according to a neural or statistical algorithm [45]),
- Can be implemented in an efficient form which greatly reduces the computational cost and storage requirements of the algorithm, in the case of a singleton input set the centroid is given by the sum of normalised membership functions multiplied by their respective weighting i.e., $y = \sum \mu_A^*(x) w_j$. For a discrete fuzzy distribution the defuzzified output is a weighted average of the fuzzy singleton outputs over the support[46] of the fuzzy input set $\mu_A(x)$.

4.2.7 Optimisation of the fuzzy algorithm for real time processing

The previous subsections have briefly alluded to issues regarding implementation of a fuzzy controller in a real-time environment. In particular individual rule firing and weighted average defuzzification were mentioned as being computationally inexpensive in comparsion to matrix based inference and other the types of defuzzification described earlier. This section will review further optimisations with respect to implementation of a controller in a real-time environment. Use of this

knowledge will be incorporated as good practice for the implementation of the controller described in this thesis.

Bolinger [47] devised a fuzzy control algorithm suitable for implementation on a Digital Signal Processor. The algorithm was based on evaluating a fixed number of rules from a rulebase of arbitrary size. The advantage of this scheme was that there is a nearly constant time of execution for each control iteration and that all rules do not have to be evaluated in order to derive a control output . The optimisation was subject to a number of constraints:-

- Only triangular membership sets were used for the antecedants - this decision was made based on the observations that the triangular set is easy to visualise, compute and store in memory, triangular sets are widely used in control applications, and the benefits derived from adjusting the system behaviour through use of different set shapes has only a minor effect on the final output.
- Only singleton membership sets are used for the output variables - the choice of singletons has the advantage of not requiring the computation of a degree of membership for a particular set a much smaller storage requirement.
- Weighted average defuzzification is used - as described in a previous section this technique can be used for defuzzification of controllers that have singleton membership sets for the output variables.
- The $t-norm$ or min operator was used for inference
- for inference the MIN operator satisfies the $T-2$ norm or multiplication - this is similar to inference in the individual rule based scheme yielding a scaled output set.
- For a given input value the sum of membership in all sets is normalised - the effect of this constraint is to eliminate a divide operation from the equations and reduces the computation of defuzzification to linear interpolation between points

Surmann and Ungering [35] presented an optimised fuzzy algorithm based on lookup tables, optimised rule processing and digital pulse duration modulation. The algorithm was suitable for implementation on general purpose processors. Their approach is summarised below:-

- Use of integer for all computations - since integer calculations are generally an order or two faster than with floating points.
- Storing the membership functions in lookup tables so that the antecedants are calculated very quickly. Their implementation required a 1Kbyte memory block per fuzzy set for a 10 bit input resolution and a 8 bit internal resolution.
- Using the most suitable algorithm for the digital processor - the Togai's Fuzzy Centre Of Gravity (FCOG) method which calculates the centre of area and area of a fuzzy set before runtime was used so that the only addition and multiplication was needed in the defuzzification stage. It was noted that standard microprocessors do not have instructions to compute minimum or maximum so several machine instructions are required for this computation. For lower I/O resolutions (less than 10 bits) these operations could be implemented via standard AND and OR logic gates. In order to take advantage of this configuration it is necessary to convert the representation of the membership functions to a pulse-duration-modulation (PDM) representation. Further speed up was achieved by use of output singleton values where the areas of the output sets are normalised.
- Including data dependencies to minimise the number of calculations in the inference process - here rules that were not activated by an input combination were not evaluated further. This jump optimisation scheme was found to considerably reduce the computation time of the inference process.

Boverie *et al.* [48] did a comparative study with regard to robustness of fuzzy control and conventional control approaches such as optimal and PID in the case of a linear dynamic process. They

also examined the sensitivity of the fuzzy controller to the design parameters. In the design of the fuzzy controller they considered the following:-

- Sampling frequency - they stressed that the controller sampling must verify Shannon's sampling theorem i.e., having a sample rate of at least twice the frequency of the controlled plant, in order to prevent high overshoot and oscillation in response to a step input.
- The quantisation of the input fuzzy sets (look-up table approach) was found to influence the accuracy, overshoot and damping effects of the system - fine quantisation was found to generate an overshoot whereas coarse quantisation favoured damping.
- Overlap and shape of fuzzy sets - they found that the form of the fuzzy sets chosen symmetrically about their maximum value, did not have great influence on the control quality (they compared, gaussian, square and triangular sets) although they found that the overlap ratio, T_x[1], did. An optimum value of $T_x = 1$ was required for best control quality in terms of overshoot and rise-time (corresponding to a 0.5 point of intersection for symmetrical sets).
- Number of rules - for an optimised rulebase (in the case when system expertise is perfect) they found that the number of rules did not affect control performance.
- Different methods of defuzzification did not have a large impact on control performance and were merely considered as a factor for fine-tuning the control. The max norm yielded a slightly higher overshoot than then CoG and mean of maxima which yielded a critically damped and slightly underdamped response respectively.

In their comparison of controllers they found that fuzzy control gave a better response with respect to response time and overshoot. They also found that the maximum value of the control was lower than PID by 25% and lower than optimal control by 60%. Fuzzy control gave the best control response for a system with modelling errors considering neglected dynamics, internal disturbances, disturbed control, disturbed output and energy consumption. Boverie *et al.* noted that PID control gave the best response when considering pure time delays. They indicated that it was necessary to develop a fuzzy predictive model or combination of a PID and fuzzy controller in order to give a better performance when there are pure delays in the system.

Brown and Harris [49] compared the use of output set and singletons in a fuzzy system. They found that the use of fuzzy sets only really has advantages in systems where a fuzzy model needs to be inverted or if the output sets form inputs to another fuzzy network. Otherwise, the use of fuzzy output sets was more computationally expensive and in the case of a learning system required the formulation of more complex learning rules in order to store the same information. These results were derived for fuzzy networks which use algebraic operators[2] which are favoured for real time implementations as reviewed earlier in this section.

Costa et al. [36] evaluated different hardware solutions for real-time fuzzy control. They divided their study into four different categories, viz.,

- Software and hardware solutions with general purpose components.
- General purpose processors with instructions for specialised fuzzy computations.
- Dedicated fuzzy co-processors.
- Fuzzy ASICs for standalone implementations.

Their study showed the specialised ASIC implementations to have the quickest system response time (order of nano seconds) but to be the least flexible as the implementation was application specific more hardware would be required for the non-fuzzy processing. The dedicated fuzzy co-processors were slightly more flexible with limited rule and membership function changes but a little

[1] $= e(N-1)\mu/L$ where e, N, L, and μ are function width, number of functions, number of quantisation values and intersection point between 2 functions

[2] as opposed to *truncation* operators such as the min intersection used in Mamdani inference which results in a clipped set

slower (order of μ seconds), this solution also required separate hardware for non-fuzzy processing. The general purpose processors with specialised fuzzy instructions (GPFI) had system response times in the order of milli-seconds and were completely flexible in terms of applicability and choice of fuzzy controller structure. The off-shelf general purpose processor had slightly slower system response times (order of seconds to milli-seconds) than the former implementation (although top of the range 32-bit processors could match the speed of the GPFI) and had the additional advantage of having a lower cost.

4.2.8 Related research by other workers

This section describes work carried out by other researchers on error based controllers fuzzy controllers that has bearing on the design of the controller in this thesis.

4.2.9 Previous work on the Middlesex prototype optically guided robot - Choice of the lower level controller

In the design proper, not only does the topology have to be selected but also the underlying lower level controller needs to be defined. In the following sections a review of related work carried out on the prototype robot by other workers is used to justify the choice of the lower level controller as well as give further insight into the selection of controller parameters including input and output scaling, fuzzification and defuzzification methods, rule base and membership function construction and their representation.

There has been ongoing development on the prototype robot at Middlesex University over the past five years (a successful PhD, two MSc's and five undergraduate projects concerned with various aspects of study on the prototype robot). Four of the previous projects addressed the problem of designing a controller for the arm. Different aspects were looked at, including the design and implementation of the controller for the positioning head [50], measurement of the robot parameters [51], implementation of different classical controllers including P, PD and PID for accurate tip placement of the tip of the flexible manipulator in the optically guided configuration [52][53][54] and design of an optimal controller for the manipulator (simulation) [55].

Study of classical controllers

Lewis [54] studied the behaviour of different classical controllers in a bid to find the controller best suited for the optically guided long reach robot. P, PD and PID controllers were compared. Lewis found that the two and three term controllers, i.e., PD and PID, performed better in comparison to single-term P control. PD control was shown to give the best rise-time and settling time and PID the best steady state error. Further experiments were carried out by Lewis to examine the tracking response of the controller over a series of steps, as would be experienced during movement of the arm 30° upwards from a horizontal position. Fig. 4.7 depicts the tracking response results obtained by Lewis. Processing of the control algorithm was carried out on a PC based intel 486 running at 25 MHz. Signal i/o was performed using a PC30B card [56] with a maximum sampling rate of 30kHz and the control software was written in C.

The response with the integral term is shown to yield smaller (mean) tracking and steady state errors in comparison to the PD control algorithm. However, Lewis noted that the effect of integral the integral term resulted in 'bump' due to integral windup and a considerable increase in settling times. To enable simultaneous deflection compensation and tracking as under normal robot operation, Lewis proposed PD control which produced a satisfactory response in both cases.

It can be seen that both algorithms have a saw tooth like waveform while tracking. This occurs because the control algorithm is constantly trying to minimise errors at the tip due to the command steps. As the error is closed the next command step is issued. The amplitude of the fluctuation from the mean tracking error for both controllers is also the same. Smoother motion can be achieved by re-design of the positioning head. A continuous command signal rather than a stepped one that is moving at a rate that the tip can track would be desirable for smooth tracking. Alternatively smoother motion and more faithful tracking could be achieved by making the tip respond quicker

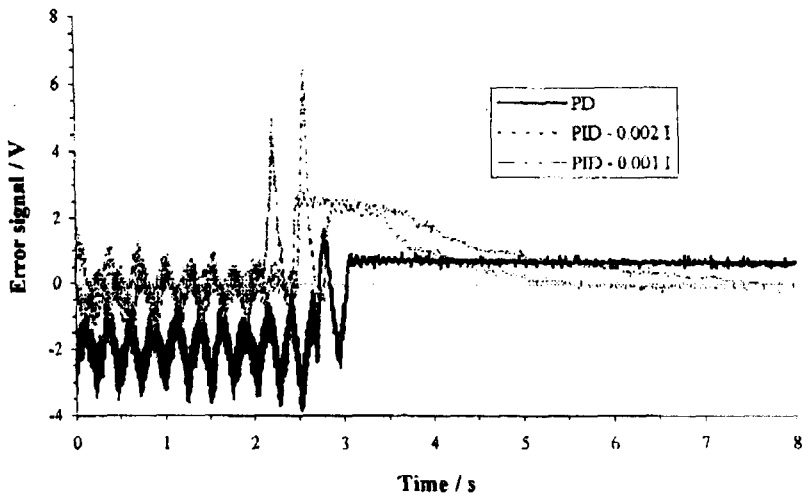

Figure 4.7: Tracking response for PD and PID control

to the generated tip tracking error. The former solution requires a hardware change and the latter may not be achievable as it is dependent on the system bandwidth.

In using the classical control approach a best compromise is sought i.e., the problem of controlling the end-point of the flexible manipulator can be divided into two separate tasks [57]:-

- the *regulation problem* - here the aim is to keep the end-point of the manipulator as close as possible to the set-point as defined by the positioning head in the presence of disturbances which act on the system or measurement,
- the *servo* or *tracking problem* - the aim is to let the end-point of the manipulator follow as closely as possible the trajectory defined by the positioning head which comprises of a series of 'steps'.

It would be desirable if the controller were able to modify its parameters to the present task rather than find a compromise solution as seen in classical control. This suggests a form of non-linear adaptive control is required. One way to make the system adaptive, without radically changing the controller algorithm, would be to use a fuzzy supervisor as described earlier in this section. These results also support the selection of a PD controller as the lowest level controller in the development of a fuzzy based controller described in this thesis.

Study of an optimal controller

Jeyadeva [55] investigated the development of an optimal Linear Quadratic Regulator (LQR) for the optically guided arm. An approach similar to that of Cannon and Schmitz [58] was adopted, except that no noise terms (Gaussian) were included in the design. The initial model adopted for the design of the controller was a linear version of that developed in the chapter on modelling in this thesis. However, using MATLAB to analyse the Jacobian (derived in ACSL) showed states of the model required in the controller design to be both uncontrollable and unobservable. This led to the decision of identifying the model experimentally. The state-space representation of the output of the flexible system is given as [59]:-

$$y = C(sI - A)^{-1}Bu \tag{4.23}$$

For sensors located at the hub and the tip this reduces to

$$\frac{\dot{\theta}}{T} = \frac{1}{I_T s} + \frac{1}{I_T}\sum_{i=1}^{\infty}\left[\frac{d\phi}{dx}(0)\right]^2 \frac{s}{s^2 + 2\zeta_i\omega_i s + \omega_i^2} \tag{4.24}$$

$$\frac{y_T}{T} = \frac{L}{I_T s^2} + \frac{1}{I_T} \sum_{i=1}^{\infty} \left[\frac{d\phi}{dx}(0)\right]^2 \frac{\phi_i(L)\frac{d\phi}{dx}I(0)}{s^2 + 2\zeta_i\omega_i s + \omega_i^2} \tag{4.25}$$

The parameters for Eq. 4.23 including the circular frequencies ω_i, modal damping ζ_i and residues $\left[\frac{d\phi}{dx}(0)\right]^2$ were calculated from the experimental frequency response curves described in Chapter 3. This model proved to be controllable but not observable. A full-order observer designed via pole placement was used to estimate the unobserved states. MATLAB was used to solve the optimal control problem which minimises a quadratic performance index that satisfies Liapunov's second method [60]. Fig. 4.8 shows the simulated step response with the controller tuned for no overshoot. It can be seen that the step response although inherently stable[3] and critically damped is much slower compared to the classical PD controller of Lewis which had a settling time of 0.3s [54].

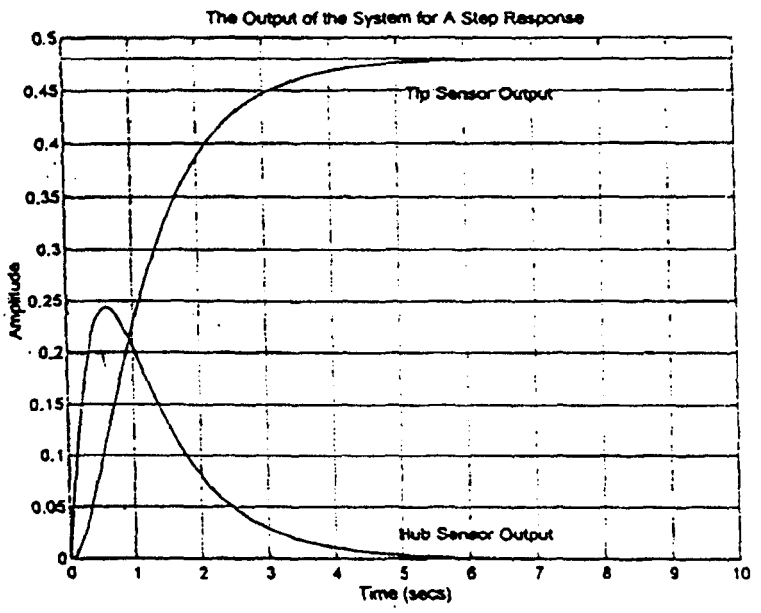

Figure 4.8: Step response under optimal control

Jeyadeva tried to improve the speed of the response by altering the pole placement margin and weighting of the Q and R matrices that define the quadratic cost function. Further tuning and a pole placement margin of ten times the state feedback resulted in an improvement of approximately 10% in rise-time this is still an order of magnitude slower than the classical PD controller.

The implications of these investigations are that it is possible to attain optimal control of the optically guided robot in the TFS based configuration. Satisfaction of the stability constraint placed on the controller could be a possible reason for the relatively slower response. Other factors in the design, such as choice of the performance index and accuracy of the model, could also influence the controller's response.

The solution of the optimal control problem via pole-placement requires an accurate model of the system. In the simulation above, identification was carried out empirically for the prevailing conditions of the robot. This method of design and control would not suffice for systems whose parameters are likely to change, for example when different payloads are to be handled by the manipulator. A method of online identification of the system parameters [61][62] or a controller that has intrinsic knowledge about the variation of the system parameters would be required in this case. From the results above it can be seen that classical PD control which is relatively simple and quick to implement provides a suitably quick step response in comparison to optimal LQR control.

Foo [63] carried on with Jeyadeva's work and devised an optimal controller based on output state feedback (LQRY controller). Tip position and hub position were used as the output feedback states in his controller. Using this scheme Foo showed that it was possible to obtain much faster settling

[3]stability is implied in the design of the LQR controller as it must conform with Liapunov's second stability criterion

times (over ten times) than for LQR control. Although the transient response exhibited slight overshoot (5%) the settling time, T_s, was 0.4 seconds. This settling time compared well with the figures achieved by Lewis using classical PD control on an unloaded arm (0.3 seconds). Although Foo's optimal results are impressive, his controller did not consider the change in dynamics due to different applied payloads and other non-linear factors such as gear backlash and coulomb friction that the arm is inevitably subject to. In order to cope with payload change and non-linear dynamics, the linear model investigated by both Foo and Jeyadeva would require some means of on-line identification such as the investigated by Rovner and Cannon [61]. These complex sophisticated techniques often require powerful processing capability and the problem becomes compounded when out of plane vibrations are to be considered and a further axis controlled.

Wang [64] compared optimal and non optimal control strategies for a single flexible link. He showed that use of flexible body LQR controller[4], H_∞ and bounded-input H_2 optimal control strategies gave a quicker step response in comparison to passive single or second order flexible control, such as the controller and model used in [65]. However, he noted that the optimal control methods were not essentially a better choice as the controllers, in an effort to reduce either a quadratic cost function or L_2 norm of the signal, resulted in large torques and large initial deflections which could easily cause mechanical fatigue of the physical system to occur. The order of the optimal controllers was also close to the order of the model considered (seventh order for two flexible modes) implying difficult implementation in terms of requiring high sampling frequencies and large computational overhead. From these studies Wang's recommendation was that controllers which are simpler in structure, robust and stable whilst maintaining low overall elastic deformations during the movements of the arm, should be investigated.

4.3 The structure of fuzzy controllers

Since the first fuzzy controller applied to the control of a steam engine by Mamdani in 1971, fuzzy control has been applied in numerous applications. Extensive application reviews are given in [66] and [67]. The plethora of applications is diverse, ranging from medical to industrial to domestic applications and the methods of implementation also vary. Use of fuzzy control is very much an application driven and although the underlying analysis of such controllers is not as well understood as conventional control methods they continue to be used extensively. This is because they are found to perform well and in some cases outperform conventional controllers.

Generally fuzzy expert control can be applied in two ways. Either as a direct controller or in a supervisory capacity [19][2]. In direct fuzzy (DF) control, the fuzzy system comprises of the main controller in the system and the rule base determines the control actions. In contrast, a supervising fuzzy (SF) controller is used to modify the parameters of an existing controller. The structure for both DF and SF controllers is depicted in Figs. 4.9 and 4.10 respectively.

There are no hard and fast rules as to which structure of controller should be adopted for a particular control problem. Generally SF control can be applied in a situation where a lower level controller exists and can be used either to switch between controllers for example in [68] or to tune the parameters of a controller according to the prevailing plant operating conditions. DF controllers on the other hand are more flexible and do not need a lower level controller. The fuzzy rules are applied directly to the plant.

The stability of the SF controller is dependent on the lower level controller since SF control merely acts as a gain-scheduling or parameter-tuning algorithm, although care must be taken when making the appropriate controller selection or parameter adjustment. For the DF controller analysis proves to be a little more difficult. This is reflected in the relatively slow acceptance of fuzzy control into the control engineer's toolbox.

Current research efforts are directed towards devising methods of producing design methods for stable analytically tractable controllers with varying degrees of success. Palm [69] has shown that fuzzy control can outperform sub-optimal switching control and has devised a recipe for a 'fuzzy switching' controller - essentially this is a rule-based implementation of sliding mode control with

[4]where the model takes in account the flexibility of the link

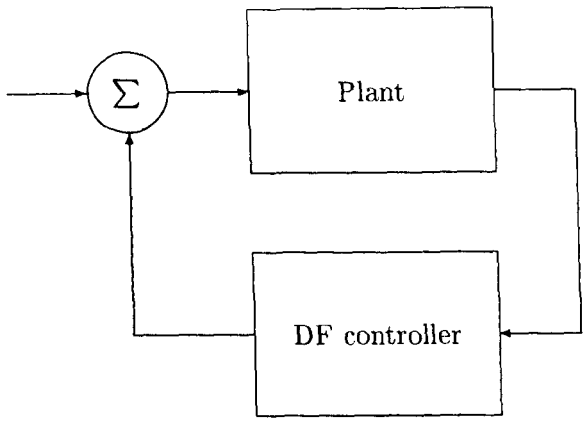

Figure 4.9: Structure of DF controller [19].

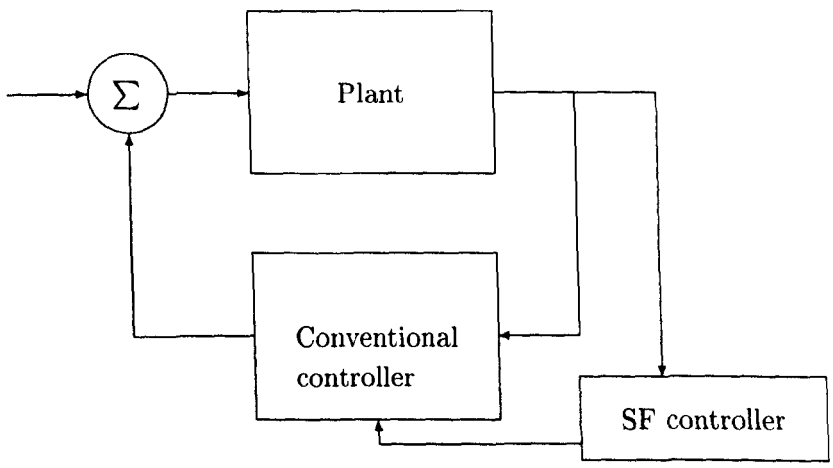

Figure 4.10: Structure of SF controller [19].

the additional flexibility provided by one's ability to heuristically tune the rules although there are strict constraints on the choice of rules. Harris *et al.* [70] have investigated the phase plane technique for error-based controllers. Their methods can be used for both the analysis and design of fuzzy rules.

For the fuzzy controller described in this thesis the decision of which structure to use must be resolved. The SF control structure has been chosen for a number of reasons. These are outlined in the following section.

Reasons for SF control

Safety Although the design of a DF controller strives to avoid this, there may be situations when the plant behaves unpredictably and the fuzzy controller yields a spurious or no output (no rules fired or badly designed rules). The spurious output can be handled by additional exception handling rules in both SF and DF controllers. In the case where there is no output due to no rules being fired it would be desirable to have some form of control over the plant behaviour. In the case of a SF fuzzy controller the lower level control will be still operating. Brown recommends that a neurofuzzy PID controller be initialised from a linear controller thus ensuring that the fuzzy controller is at least as good as the linear controller [71]. Passino and Yurkovich [72] note that there are several disadvantages to using DF control where the

fuzzy experts rules are determined heuristically. They state that the following areas beg questions about derivation of a rule base that are cause for concern especially in safety critical environments:-

- Is artisan knowledge able to include all the situations that can occur due to disturbances, noise or plant parameter variations?
- Can the human expert realistically and reliably foresee problems that could arise from closed-loop system instabilities or limit cycles?
- Will the expert be able to effectively incorporate stability criteria and performance objectives into a rule base in order to ensure that reliable operation can be obtained?
- Can an effective and widely used synthesis procedure be devoid of mathematical modelling and subsequent use of proven mathematical tools?

Although much research activity is being carried out in DF control, the above questions are yet to be answered. Ewers and Bordeneuve-Guibe [73] successfully applied a fuzzy supervisor about a conventional controller to an anti-skid system for a commercial aircraft and a security supervising system for the longitudinal control of a cargo aircraft. They used this approach to *"ensure a safe, though optimal performance."*

The approach in this thesis is to follow the safe path by extending the capability of a well known analytically tractable controller rather than devising a pure DF controller which invariably relies on an experts knowledge.

Extension of the performance envelope of the lower level classical controller By varying the parameters of the classical controller online, it is possible to make the controller adaptive and thereby cope with control problems of a non-linear nature. Traditionally classical controllers were tuned for a system under a particular configuration and any change in parameters would cause the tuning to be non-optimal. Adding 'intelligence' to the classical controller should also lead to a more robust system that can cope with uncertainty in the system. Beneficial operator knowledge can also be applied to the controller.

Shorter development time Designing a SF controller about an existing controller will have a shorter development time over a DF based controller. The reason for this is that with an existing controller there is already something tangible to experiment with in terms of deriving rules and testing their effects on overall control performance. A DF controller on the other hand would be less easy to design without any apriori knowledge of the working of the system and there is a large step to be made from 'no control' to 'DF control'. Yurkovich recommends the design of a fuzzy controller via the SF approach if it is available before moving on to more complex designs [74].

Basis for Evaluation The use of supervisor based configuration has the advantage of having the linear classical system form a basis for the stability analysis of the control system with the fuzzy controller setting the controller limits. Also, the classical controller can be used directly as a benchmark to evaluate the performance of the fuzzy based controller.

Acceptance in Industry Most industrial controllers to date still employ the well understood classical PID control. The reasons for this are in the underlying theory concerning the design of such controllers and over forty years of successful use. These controllers are relatively simple to analyse and in many cases industry is unwilling to pay for the installation and maintenance of little known highly sophisticated controllers. The approach of adding intelligence to an already well known controller could be more attractive to industry financially and legacy-wise. Brown and Visavadia [71] state that fuzzy PID controllers, in the DF configuration, have been received with scepticism in the UK despite being developed here since there are no formalised methods for their design and analysis. To date the popularity of fuzzy controllers has been mainly application driven [66].

Previous applications using a similar controller Hierarchical control of flexible manipulators using fuzzy logic has been successfully demonstrated by [75]and [68] as reviewed in Chapter 2 of this thesis.

The difference between self-tuning and fuzzy supervised control Jacketing a classical controller with a fuzzy controller is in many ways similar to adaptive self tuning control in that the control parameters are adjusted according to the dynamic behaviour of the plant [72]. The difference is in that fuzzy control is better suited to dealing with uncertainty in a system and more specifically does not require a plant model in order to derive the next control law.

4.4 Design recommendations for the fuzzy controller - Discussion

This chapter establishes the theory and structure for a real-time fuzzy controller that is suitable for the TFS based application described in this thesis.

For a real time implementation the following features of the fuzzy controller are desirable:-

- Individual rule based inference.
- Lookup table implementation of the fuzzy antecedants.
- Straight edged fuzzy sets with a 0.5 crosspoint or 25 % overlap.
- Use of *min* for the intersection norm in the inference procedure, multiplication for firing the individual rules and sum for composition.
- Integer based calculations.
- An adequate discretisation of the fuzzy sets in order to preserve the linguistic representation of the fuzzy set and to prevent the effects of aliasing.
- Evaluation of a fixed number of rules from an arbitrary rule base in order to achieve a uniform sampling rates.
- Singleton vectors as opposed to output sets in the consequent.
- Weighted average defuzzification.

Previous research into the control of the TFS based manipulator has shown the classical PD control gives the best overall performance in comparison to other classical methods and optimal control. For this reason PD control is chosen as the lower level in the SF based controller. This configuration is chosen or the following reasons:-

- It is a cautious approach to fuzzy controller design - this is desirable since the application is to be deployed in a safety critical environment.
- The performance envelope of the PD classical controller can be extended and made more robust.
- Considerable knowledge exists on the tuning of traditional controller which can incorporated in the fuzzy rules.
- Evidence of previous successful applications of SF based controllers but with different lower level controllers on similar systems.
- Shorter development time and simplicity of the method.
- The availability of analysis tools for the lower level controller - the supervisor sets the gain bounds of the classical controller and as long as the locus of the controlled system under PD control is stable within these bounds the overall controller will be stable.

4.5 Summary

This chapter has evaluated fuzzy control in the context of applying it to the TFS based robot in a safety critical environment. The choice of the structure of the fuzzy controller has also been investigated. A supervising fuzzy controller with a lower level PD controller has been recommended based on research by other workers into the classical and optimal control of the TFS based robot.

Criteria for the topology for a real-time implementation of the fuzzy controller have also been investigated. These have been outlined in section 4.4. The choices for the fuzzy design parameters or topology have been made with the intention of achieving an algorithm that is both memory efficient and can be implemented in real-time but at the same time provides fidelity in the representation of the control rules.

References

[1] Pfeiffer B. and Isermann R. Criteria for successful applications of fuzzy control. *Engng. Applic. Artif. Intell.*, 7(3):245–253, 1994.

[2] Verbruggen H. B., Krijgsman A. J., and Bruijn P. M. Towards intelligent control: Integration of AI in control. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of artificial intelligence in process control*, pages 223–249. Pergamon Press, 1992.

[3] Tzafestas S. AI techniques in control:An overview. In *Int. Symposium on AI, Expert Systems and Languages in Modelling and Simulation*, 1987.

[4] Albertos P. Fuzzy controllers. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of Artificial Intelligence in Process Control*, pages 343–367. Pergamon Press, 1992.

[5] Kosko B. *Neural Networks and Fuzzy Systems*, chapter 1, page 19. Prentice Hall, UK, 1992.

[6] Kosko B. *Neural Networks and Fuzzy Systems.* Prentice Hall, UK, 1992.

[7] Boullart L. A gentle introduction to artificial intelligence. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of artificial intelligence in process control*, pages 5–31. Pergamon Press, 1992.

[8] Patton R., Frank P., and Clark R. *Fault diagnosis in dynamic systems. Theory and applications.* Prentice Hall, New York, 1989.

[9] Kuipers B. Qualitative reasoning. *Automatica*, 25(4):571–585, 1989.

[10] Soeterboek A. R. M. *Predictive control - A unified approach.* Prentice Hall, 1992.

[11] Ritter H., Martinetz T., and Shulten K. *Neural Computation and Self Organising Maps.* Addison-Wesley, 1992.

[12] Lippmann R. P. An introduction to computing with neural networks. *IEEE ASSP Magazine*, pages 4–22, 1987.

[13] Khanna T. *Foundations of neural networks.* Addison Wesley, USA, 1990.

[14] Kung S. Y. *Digital neural networks.* Prentice Hall, USA, 1993.

[15] Hunt K. J., Sbarbaro D., Zbikowski R., and Gawthrop P. J. Neural networks for control systems - A survey. *Automatica*, 28(6):1083–1112, 1988.

[16] Miller W. T., Sutton R. S., and Werbos P. J., editors. *Neural Networks for control.* MIT Press, 1990.

[17] Wells G. A introduction to neural networks. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of artificial intelligence in process control*, chapter Part I : Artificial Intelligence, pages 164–200. Pergamon Press, 1992.

[18] Pegman G. Safety and standards for advanced robots. In *IEE Colloquium on Safety and Reliability of Complex Robotic Systems*, Savoy Place, London, 1994.

[19] Drainkov D., Hellendoor H., and Reinfrank M. *Introduction to Fuzzy Control.* Springer Verlag, 1993.

[20] Zadeh L. A. Fuzzy Algorithms. *Information and Control*, 12:94–102, 1968.

[21] Zadeh L. A. A Rationale for Fuzzy Control. *J. Dynamic Systems, Measurement and Control*, 94:3–4, 1972.

[22] Zadeh L. A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man, and Cybernetics*, SMC-3:28–44, 1973.

[23] Black M. Vagueness : An exercise in logical analysis. *Philos. Sci.*, 4:427–455, 1937.

[24] Vepa R. Introduction to fuzzy logic and fuzzy sets. In Boullart L., Krijgsman A., and Vingerhoeds R. A., editors, *Application of artificial intelligence in process control*, chapter Part I : Artificial Intelligence, pages 146–163. Pergamon Press, 1992.

[25] Thomas D. E. and Armstrong-Helouvry. Fuzzy logic control - A taxonomy of demonstrated benefits. *Proceeedings of the IEEE*, 83(3):407–421, 1995.

[26] Dubois D. and Prade H., editors. *Fuzzy sets and systems: Theory and applications.* Academic Press, Orlando FL, 1980.

[27] Zadeh L. A. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.

[28] Weber S. Ageneral concept of fuzzy connectives, negations and implications based on *t-norms* and *t-co-norms*. *Fuzzy Sets and Systems*, 11:115–134, 1983.

[29] Zadeh L. A. Calculus of fuzzy restrictions. In Zadeh L. A. et al., editor, *Fuzzy sets and their applications to cognitive and decision processes*, pages 1–39. 1975.

[30] Zadeh L. A. The concept of a linguistic variable and its application to approximate reasoning - part i. *Information Science*, 8:199–249, 1975.

[31] Zadeh L. A. The concept of a linguistic variable and its application to approximate reasoning - part ii. *Information Science*, 8:301–357, 1975.

[32] Zadeh L. A. The concept of a linguistic variable and its application to approximate reasoning - part iii. *Information Science*, 9:43–80, 1975.

[33] Mamdani E. H. Application of fuzzy algorithm for control of simple dynamic plant. *Proc. IEEE*, 121(12):1585–1888, 1974.

[34] Surdhar J. S., White A. S., Gill R., and Mistry G. An efficient fuzzy algorithm applied to a flexible link 1dof robot. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 68–72, New Orleans, Louisiana, USA, 1996.

[35] Surmann H. and Ungering A. P. Fuzzy rule based systems on general purpose processors. *IEEE Micro*, pages 40–48, 1995.

[36] Costa A., De Gloria A., Farabischi P., Pagni A., and Rizzotto G. Hardware solutions for fuzzy control. *Proceedings of the IEEE*, 83(3):422–434, 1995.

[37] Kosko B. *Neural Networks and Fuzzy Systems*, chapter 11, page 385. Prentice Hall, UK, 1992.

[38] Kosko B. *Neural Networks and Fuzzy Systems*, chapter 8, page 314. Prentice Hall, UK, 1992.

[39] Brown M., Mills D.J., and Harris C.J. The representation of fuzzy algorithms used in adaptive modelling and control schemes. *Fuzzy Sets and Systems*, 79:69–91, 1996.

[40] Takagi T. and Sugeno M. Derivation of fuzzy control rules from human operators control actions. In *In Proc. IFAC symposium on fuzzy information, knowledge representations and decision analysis*, pages 55–60, 1983.

[41] Jang R. J. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685, 1993.

[42] Tsukamoto Y. An approach to fuzzy reasoning method. In Gupta M.M., Ragade R.K., and Yager R. R., editors, *Set Theory and Applications*, pages 137–149. North-Holland, 1979.

[43] Lee C.C. Fuzzy logic in control systems: Fuzzy logic controller - Part i. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):405–417, 1990.

[44] Lee C.C. Fuzzy logic in control systems: Fuzzy logic controller - Part ii. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):419–435, 1990.

[45] Kosko B. *Neural Networks and Fuzzy Systems*, chapter 1, page 29. Prentice Hall, UK, 1992.

[46] Drainkov D., Hellendoor H., and Reinfrank M. *The mathematics of fuzzy control*, chapter 1, page 52. Prentice Hall, UK, 1993.

[47] Bolinger E. Optimisation of the fuzzy control algorithm. *Information sciences - Applications*, 2(3):135–142, 1994.

[48] Boverie S., Demaya B., and Titli A. Fuzzy logic control compared with other control approaches. In *Proceeedings of the* 30^{th} *conference on decision and control*, pages 1212–1216, Brighton, England, 1991.

[49] Brown M. and Harris C. J. Fuzzy output sets: Their (mis)use in modelling and control. In *IEE colloquium on 2 decades of fuzzy logic*, pages 6/1– 6/5, IEE Savoy Place London, 1993.

[50] Petit B. The analysis, development and testing of aspects of a laser guided robot control system,Erasmus project, (BSc industrial engineering - Nuclear option, Institute Gramme Leige Belgium), Middlesex University, London, England. 1994.

[51] Pape M. An investigation of the structural and dynamic properties of an optically guided arm, Erasmus project, Middlesex University, London, England. 1995.

[52] Lewis J. The design, construction and testing of a laser guided robot arm, MSc dissertation, Middlesex University, London, England. 1991.

[53] Korhonen J. The control and testing of a laser guided robotic arm,MSc in Applied Computing Technology and Computers in Industry, Middlesex University, London, England. 1995.

[54] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[55] Jeyadeva J. Flexible robot control, Final year project - BEng. (Hons) in Mechatronics Engineering, Middlesex University, London, England. 1997.

[56] Editor. User manual for PC-30B/C/D. Technical report, Amplicon Liveline Limited, Centenary Industrial Estate, Brighton, East Sussex BN2 4AW, 1990.

[57] De Scutter J. An introduction to PID control and its application to motion control. In *Lecture notes of the short course on computer controlled motion - Third Edition*, pages 79–112. Katholieke Universiteit Leuven, Celestijnenlaan 300 B - 3001 Heverlee - Belgium, 1992.

[58] Cannon R. H. and Schmitz E. Initial experiments on the end-point control of a flexible one-link robot. *The International Journal of Robotics Research*, 3(3):62–75, 1984.

[59] Schwarzenbach J. and Gill K. F. *System modelling and control.* Edward Arnold a division of Hodder and Stoughton, 1992.

[60] Schwarzenbach J. and Gill K. F. *System modelling and control*, chapter 12, page 269. Edward Arnold a division of Hodder and Stoughton, 1992.

[61] Rovner D. M. and Cannon R. H. Jr. Experiments toward on-line identification and control of a very flexible one- link manipulator. *The International Journal of Robotics Research*, 6(3):3–19, 1987.

[62] Alder J. and Rock S. M. Experiments in control of a flexible-link robot manipulator with unknown payload dynamics:An adaptive approach. *The International Journal of Robotics Research*, 13(6):481–495, 1994.

[63] Foo S. S. Flexible Robot Controller, Final year project - BEng. (Hons) Mechatronics, Middlesex University, London, England. 1998.

[64] Wang D. Comparison of optimal and non-optimal control strategies for the single flexible link. *International journal of robotics and automation*, 9(3):130–136, 1994.

[65] Wang D. and Vidyasagar M. Transfer functions for a single flexible link. *IEEE Trans. on Robotics and Automation*, 5(3):373–377, 1989.

[66] Maiers J. and Sherif Y.S. Applications of fuzzy set theory. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):175–189, 1985.

[67] Rudall B. H. Fuzzy Control Applications. *Automatica*, 9:126–132, 1991.

[68] Moudgal V. G., Kwong W. A., Passino K. A., and Yurkovich S. Fuzzy learning control for a flexible link robot. *IEEE Transactions on Fuzzy Systems*, 3(2):199–210, 1995.

[69] Palm R. Sliding mode fuzzy control. In *Proceedings of the IEEE* 1^{st} *International Conference on Fuzzy systems*, pages 519–526, 1992.

[70] Harris C. J. and Moore C. G. Phase plane analysis tools for a class of fuzzy control systems. In 1^{st} *Int. Conf. On Fuzzy Systems*, pages 511–518, 1992.

[71] Brown M. Q. and Visavadia K. An analysis of the capabilities of fuzzy PID controllers. In *IEE Colloquium on Fuzzy Logic Controllers in Practice*, Savoy Place, London, 1996.

[72] Passino K.M. and Yurkovich S. Fuzzy Control. In Levine W.S., editor, *The control handbook*, pages 1001–1017. CRC Press, IEEE press, 1996, 1996.

[73] Ewers B. and Guibe-Bordeneuve J. Expert supervision of conventional control systems. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 143–149, New Orleans, Louisiana, USA, 1996.

[74] Yurkovich S. A control engineers perspective on fuzzy control. In Bastin G. and Gevers M., editors, *Plenary Lectures and Mini-Courses of the European Control Conference, ECC97*, pages 75–105. European Control Conference, 1997.

[75] Garciabenitez E., Yurkovich S., and Passino K. Rule-Based supervisory control of a two link flexible manipulator. *Journal of Intelligent and Robotic Systems*, 7:195–213, 1993.

Chapter 5

Design of the fuzzy PD controller

5.1 Introduction

The previous chapter gave an overview of fuzzy theory highlighting optimisations for real-time fuzzy control. A fuzzy supervisor about a PD classical controller was chosen as the basic controller structure. The choice of this structure was based on the experience of other workers and its suitability for use in a safety critical and real-time environment.

The fuzzy design variables including the set shapes for the input and output, method of inference and defuzzification were also selected. However, the implementation specifics, such as the discretisation of the input and output sets and the justification of the fuzzy rules, still need to be addressed.

This chapter describes a methodology for the development of a variable payload fuzzy controller. Emphasis has been made on an implementation that is suitable for real-time control for a small embedded type processor where there are constraints on the amount of memory available for the application.

5.2 Fuzzy controller design

The design of a fuzzy controller can be separated into the following steps:-

Choice of controller variables In this step the state variables for the controller are determined, i.e., the controllable and observable variables of the system are identified. The behaviour of these parameters is also studied through experiments in order to establish an effective rule base for the fuzzy controller.

Scaling or normalisation Next the identified input (process state variables) and output (control output variables) parameters are mapped from a physical domain to a normalised Universe of Discourse (UoD).

Fuzzification Here the respective inputs and outputs are converted into linguistic representations via fuzzy sets.

Definition of the rulebase Based on the observations of the parameter behaviour and plant response carried out in the first step a set of rules is to define the behaviour of the controller.

Defuzzification The linguistic interpretation of the fuzzy outputs sets as a result of inference are converted back to normalised outputs or crisp outputs.

De-normalisation The output UoD is mapped back to the physical output domain.

The following sections detail the design steps as applied to the TFS based long reach robot.

5.3 Choice of controller variables

Usually in conventional control this involves identification of state variables that are observable and controllable - i.e., variables that can be measured and that can be manipulated by the controller in order to achieve the desired performance. A similar principle applies to fuzzy control. Variables must be identified that can have an impact or influence the output of the system under fuzzy control. These variables must also be realisable as linguistic fuzzy sets in order to be manipulated by the fuzzy inference engine.

The choice of input variables is constrained by the TFS based application, these are the measurement of the tip position on its rate of change, *error* and *derivative* respectively, from the laser defined set-point.

The choice of the controller output variable for the TFS based controller application depends on the choice of the lower level controller and is aided by a couple of factors. The first is that, since their inception over forty years ago, a substantial amount of research has gone into studying the behaviour of classical controllers applied to linear time invariant first and second order plants.

The influence of the different control parameters, viz., the Proportional (K_p), Derivative (K_d) and Integral (K_i) gain constants on the quality of control for linear systems is well understood [1]. From the experience of past workers on classical control it is well known that for idealised linear time invariant systems K_p influences the rise-time and percentage overshoot of the system, K_d affects the damping properties and K_i the steady state error.

The second factor is the result of experimental observation of the influence of these parameters in practice on the TFS based long reach robot. The aim of these experiments is to confirm how true these rules of thumb are with reference to the TFS based robot which is highly non-linear and has time varying dynamics. Only the proportional and derivative parameters are investigated i.e., PD control, as the integral component has been shown to yield undesirable 'bump' in the tracking performance due to integral windup [2]. The observations from these experiments are also useful in the determination of the rules for the fuzzy controller.

5.3.1 Approach to tuning

There are well established methods for the choice of classical controller parameters for first and second order linear and even first order plus dead-time systems. Included in these methods are the Zeigler-Nichols time and frequency domain methods [3], the methods of Chien *et. al.* [4] (which modified the Zeigler-Nichols parameters for better a better damped response) and the three parameter Kappa-Tau tuning methods investigated by Cohen and Coon [5]. These 'Feature-Based' techniques are often used in industry for automatic tuning since they are relatively simple to carry out and are based on an easily obtainable open-loop or closed loop step response.

A similar approach to Zeigler's second method has been used for the tuning of the PD controller applied to the TFS based robot described in this thesis. However, the method had to be modified in order to compensate for the system non-linearity due to gravity and friction effects and uncertainty in the dynamics due to different loads applied at the manipulator's tip. For motion in the horizontal plane, the method employed to determine the proportional gain constant was as follows:-

1. Under closed loop proportional control, with the arm unloaded, establish a proportional gain constant that was high enough to overcome viscous and coulomb friction in a tracking sweep for both directions of motion.

2. Increase the proportional gain constant so that there is just enough torque is available to move the arm with maximum payload.

The procedure for motion in the vertical plane was as follows:-

1. Under closed loop proportional control, with the arm unloaded, establish a proportional gain constant that was high enough to overcome gravity, i.e., provide sufficient holding torque.

2. Increase the proportional gain constant so that there is just enough torque is available to cope with maximum payload.

3. Adjust the proportional gain constant so that there is enough torque to move the arm against gravity with maximum payload.

The values determined in this way were used in experiments on the practical controller. The empirically derived gains are tabulated below. The proportional constant used in the simulations is also given.

Proportional gain value	plane of motion
0.55	vertical
0.6	horizontal
0.7	horizontal (simulation)

Table 5.1: Proportional gain constants for the simulation and practical implementation

The discrepancy between the practice and simulation can be explained by the fact that coulomb friction, motor backlash, harmonic drive flex[1] and gravity effects are not modelled. The model also assumes a uniformly thick arm and perfect mechanical linkage between the actuator and arm (in reality this is not true as the arm is constructed from a number of fixed joint components).

5.3.2 Empirical study of the effect of the derivative gain constant on control performance

Previous research into classical controllers for TFS based robot by other workers has shown that derivative action is required for a quick and damped response. Although derivative action gave desirable damping to the response, it was observed that high derivative gain caused the system to become unstable during tracking. The task becomes further complicated when different payloads are introduced at the arm's tip. Higher derivative values are required for damping the transient responses due to the additional load. This section aims to investigate the hypothetical benefits of derivative action and also aims to capture the knowledge gleaned from the observations so that it may be used in the design of a fuzzy based PD controller.

A set of experiments were conducted in order to assess the performance of the classical controller with reference to the controller parameters. Both step and tracking response were studied.

The following factors were taken into consideration:-

- Payload applied at the tip of the manipulator,
- Value of the derivative gain variable, K_d.

The measured results from step response tests using PD control in the horizontal (side-to-side movement) plane for different step directions, payload and derivative gains are given in table 5.2. The arm was positioned at a horizontal elevation with 0° yaw for these experiments.

For the tracking response, the arm was commanded to move through a 10° sweep in the right-hand-side direction. Table 5.3 below shows the tracking responses for different payloads and different derivative gain constants.

[1] although a less stiff section has been included in the model at the root of the arm to approximate the latter harmonic flex effect

K_p	K_d	$t_r(s)$	$t_s(s)$	$M_{pt}(\%)$	Payload/(Kg)
0.6	0	0.06	0.32	85	0
0.6	0	0.06	0.28	80	0.25
0.6	3	0.06	0.2	1	0.25
0.6	8	0.09	0.24	0	0.25
0.6	0	0.05	0.71	90	0.70
0.6	0	0.05	0.054	96	1.15
0.6	0	0.05	1.18	98	1.6
0.6	6	0.05	0.24	21	1.6
0.6	8	0.07	0.17	4.8	1.6

Table 5.2: Damping performance for different derivative gains and payloads

Controller	K_d	Payload/(Kg)	Stability
PD	0	0	stable
PD	0	0.7	stable
PD	0	1.15	stable
PD	0	1.6	stable
PD	4	0	stable
PD	4	0.7	stable
PD	4	1.15	stable
PD	4	1.6	unstable
PD	6	0	stable
PD	6	0.7	unstable
PD	6	1.15	unstable
PD	6	1.6	unstable

Table 5.3: Tracking performance for different derivative gains and payloads

The following observations are made:-

1. High K_d values are required for damping when there are higher payloads. $K_d = 3$ gives 1% percentage overshoot for a 0.25 kg payload and $K_d = 8$ required for a 4.8% percentage overshoot for a maximum payload of 1.6 kg.

2. The higher K_d values resulted in a slightly slower rise-times. $K_d = 8$ gives a rise-time of 0.09 s and $K_d = 0$ gives a rise time of 0.06 s for the same payload (0.25 kg). The slower rise time due to higher K_d is also observed for the max payload (1.6 kg) case.

3. Although higher derivative values give a better damping performance in the transient response , these relatively larger derivative terms($K_d = 6$) are undesirable during tracking. Table 5.3 shows that the controller becomes unstable for the higher payload cases. The instability can is possibly due to increased sensitivity of the controller to the noisy sensor signal due to the higher derivative component.

From these observations it is possible to conclude that higher derivative gain constants are desirable for damping vibrations the arm's tip. But, high K_d causes the controller to become unstable during tracking. It would be desirable to have a high K_d only during the *critical* section of the transient response where overshoots are high. Otherwise the value of K_d should be kept low so as to not compromise the controller stability during tracking.

5.4 Input scaling

The previous chapter has shown that a look-up table based on a discrete UoD is desirable for real-time fuzzy control. For a discrete implementation, a scale mapping which serves to transform measured variables into the discretised UoD is required.

Also, the choice of input resolution needs to be carefully determined as this has direct influence on controller sensitivity [6] and memory requirements. For example, if a UoD is quantised for every five units of measurement instead of every ten units, then the controller is twice as sensitive to the observed variables. The latter question is dealt with in detail in section 6.4.

The former question is more straight forward. For the controller being developed in this thesis a simple uniform (linear) mapping serves to transform the measured variables into values in the discretised universe of discourse. For fuzzy PD control the two input variables are the deviation of the tip from the laser defined set point (*error*) and its rate (*derivative*). The maximum range of the error is easily obtained from the sensor saturation curve described in Chapter 3, i.e., $max\_negative\_saturation\_voltage < error < max\_positive\_saturation\_voltage$ which corresponds to -0.003 < *error* < 0.003 radians used in the simulation. The maximum value of the derivative is obtained by differentiation of the error signal when a step input is applied. For these measurements it was important to consider all the different payloads that can be applied to the arm and the highest proportional gain for a pure Proportional controller.

Mapping to the error UoD is carried out as follows:-

$$Q = (error * K_m * N_q) + (N_q/2) \tag{5.1}$$

Where Q is the input array value, K_m a conversion constant converting the physical input domain to the input resolution and N_q is the input resolution or number of elements in the array representing the input. For a uniform mapping, $K_m = 1/2 * error$. A maximum error of $3 * 10^{-3}$ requires $K_m = 166.6$ ($N_q = 50$).

5.5 Fuzzification

The fuzzification design step is concerned with the selection of the fuzzy sets to represent the input and output data. The number of fuzzy sets in the controller has a direct influence on the number of rules (in a complete rulebase [7]). Large rule-bases result in heavy processing burdens (the inference

part of a fuzzy controller is generally the most computationally intensive [8]) and many sets have an undesirable memory overhead especially in look-up table implementations. Generally controller implementations have 3 - 7 sets as a rule of thumb[9] for both the antecedant and consequent. The width and number of the sets may vary over the UoD. Narrower width triangular sets are often found near the steady state or zero input condition where finer control is required and wider hedging sets are found in towards the edges of the UoD where more coarse control is required [10].

For the implementation described in this thesis, the number of input sets was chosen according to the rule of thumb above and by ensuring the UoD was completely covered by overlapping sets. Triangular and trapezium fuzzy sets were used. Triangular and trapezium have successfully used in many control applications [11]. The sets were designed to meet the criterion of optimal overlap ratio of $T_x = 1$ set by Bolinger. Further influence on the choice of fuzzy set that came from the data being modelled. The optical sensor transfer function can be approximated by a non-linear saturating function. This error data is best modelled by a straight edged fuzzy set.

The number of output sets was determined from the number of distinct control actions to be carried out by the fuzzy controller. These actions were discerned from the experimental and simulation analysis described in section 5.3.2 earlier. Table 5.4 outlines five levels of K_d and their effect on damping:-

K_d level	Effect on damping	Output mnemonic
$K_d = 0$	No damping	Z (Zero)
$K_d = 4$	Required for no payload	PS (Positive Small)
$K_d = 6$	Required for 0.25 kg payload	PM (Positive Medium)
$K_d = 8$	Required for 0.5 and 1.15 kg payloads	PB (Positive Big)
$K_d = 12$	Required for heaviest 1.6 kg payload	PBB (Positive Big Big)

Table 5.4: Influence of K_d on damping

5.6 Fuzzy rule design

This section deals with the design of rules for the fuzzy PD controller. Issues such as how the rules are devised from the operators observations and how they are implemented are also tackled. An analysis of an initial design leads to a design methodology for the design and implementation of more efficient rules.

5.6.1 Derivation of fuzzy rules

Fuzzy rules may be derived in a number of ways, the most common being via heuristic methods where the rules are derived from the operators observations of the plant behaviour. There are other methods of rule design based where fuzzy rules are automatically generated. Rules for these self organising fuzzy systems can be generated in a number of ways, either through artificial intelligence methods such as neural networks or genetic algorithms [12][13] or through look-up against a pre-defined performance table [14].

Rules for the TFS based application were derived heuristically from the observations of the plant and variable behaviour. This approach was adopted because there was enough behaviour data to design a rule-base. Moving onto the more complex self organising methods would be a logical progression from this approach once the basic heuristic methods are well understood [15].

The observations about plant behaviour in section 5.3.2 are used for deriving the fuzzy control rules. Rules are comprised of a combination of input antecedants which gives a rise to an output consequent that results in the desired control action. From these observations it can be seen that it would be beneficial to be able to alter the K_d gain constant according to the state of the plant. A high K_d is required for a damped step response when there is a high payload and movement is in the right-hand-side direction. However, a low K_d value is required for a stable tracking response in the same direction and for the same high payload. This implies that the controller must be

aware of not only the payload and direction of motion but also whether the motion is step-like or tracking. The latter problem was not tackled in the initial proof of concept experiments below. The rules can be coded linguistically as:-

- if e is NEGATIVE and $\dot{e}$ is NEGATIVE set K_d to a POSITIVE_MEDIUM value
- if e is ZERO and $\dot{e}$ is NEGATIVE set K_d to a POSITIVE_SMALL value
- if e is POSITIVE and $\dot{e}$ is NEGATIVE set K_d to ZERO
- if e is NEGATIVE and $\dot{e}$ is ZERO set K_d to a POSITIVE_SMALL value
- if e is ZERO and $\dot{e}$ is ZERO set K_d to ZERO
- if e is POSITIVE and $\dot{e}$ is ZERO set K_d to a POSITIVE_SMALL value
- if e is NEGATIVE and $\dot{e}$ is POSITIVE set K_d to ZERO
- if e is ZERO and $\dot{e}$ is POSITIVE set K_d to a POSITIVE_SMALL value
- if e is POSITIVE and $\dot{e}$ is POSITIVE set K_d to a POSITIVE_MEDIUM value

or in short form as:-

N,N;PM
Z,N;PS
P,N;Z
N,Z;PS
Z,Z;Z
P,Z;PS
N,P;Z
Z,P;PS
P,P;PM

These rules can be conveniently represented in the matrix in Fig. 5.1 The mnemonics P, N, S, M, E, D, Z are abbreviations for POSITIVE, NEGATIVE, SMALL, MEDIUM, ERROR, DERIVATIVE and ZERO respectively. e and $\dot{e}$ represent the input error and its derivative as measured from the optical tip sensor.

$\dot{e}$ \ e	NE	ZE	PE
PD	Z	PS	PM
ZD	PS	Z	PS
ND	PM	PS	Z

Figure 5.1: Heuristically designed fuzzy rule-base

The input space was partitioned as shown in Fig 5.2. Trapezium and triangular fuzzy sets were used to cover the respective error and derivative UoDs. Triangular sets were used for the output sets.

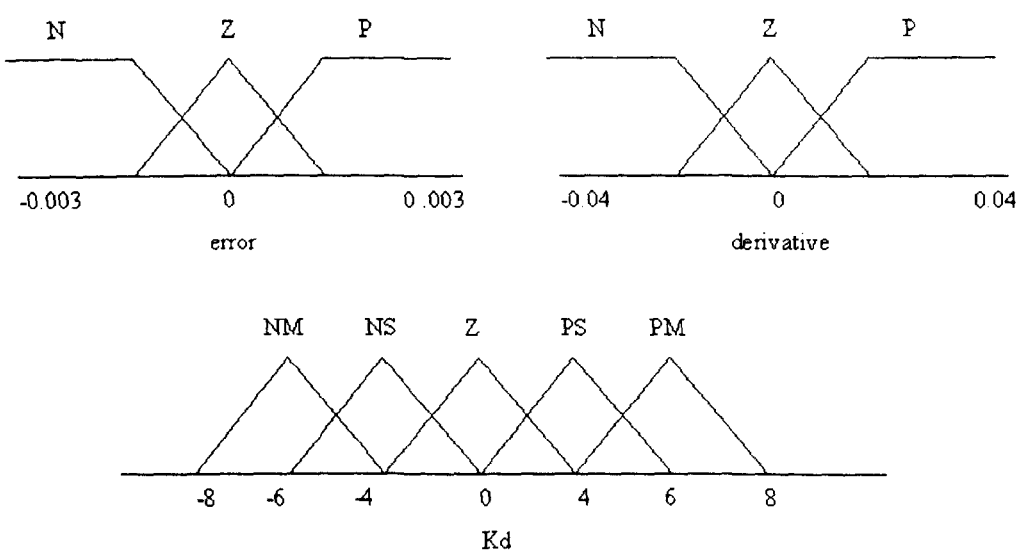

Figure 5.2: Partition of the input UoD's for the heuristically designed fuzzy controller

For the practical implementation, the fuzzy PD controller was implemented using a intel 386 processor running at 25 MHz [16]. Data acquisition was performed using a PC 30B card [17] with a sampling rate of 30 kHz. The arm remained unloaded in these experiments. The step response for the fuzzy PD controller is given in Fig. 5.3. Responses from other classical controllers are also given (NB all the classical controllers were experimentally tuned for best step response in terms of rise-time, overshoot and steady state error).

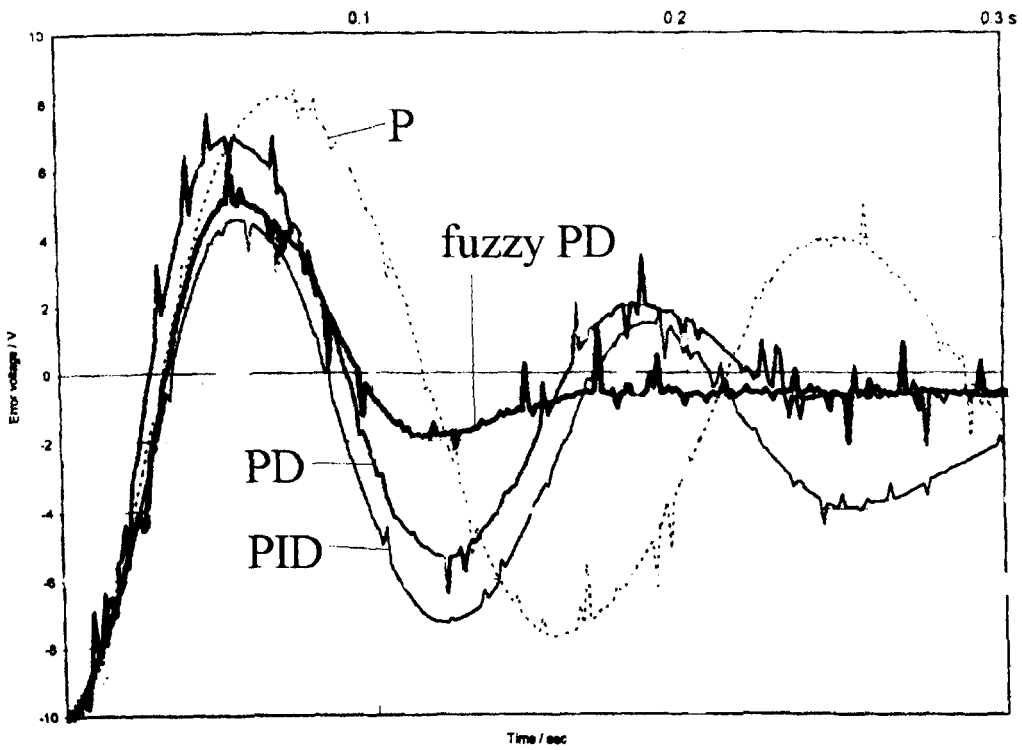

Figure 5.3: Practical step response under fuzzy PD control (serial implementation)

Fig. 5.3 shows that the fuzzy PD controller gives an improved performance over its classical counter-part. The performance is also seen to be better than the other classical control methods tested. Although the transient performance is better due to fuzzy control in comparison to other classical control techniques, the step response still leaves a lot to be desired in terms of maximum overshoot, settling time and steady state error. The response is also seen to suffer from large derivative kicks appearing on top of the average control action. The derivative kick are caused by quantisation noise in the Analogue to Digital Converter (ADC) that is amplified by the derivative term of the controller [18]. In later experiments this effect was reduced by reducing the sample rate of the derivative term (c.f. section 6.5). Improvements in the speed of processing, which influences the controller sampling rate, and more efficient rule design have been shown to improve performance significantly [19]. The following sections detail how these performance improvements are achieved.

5.6.2 Analysis of fuzzy rules

From the preliminary results above it can be seen that a set of nine rules are used to improve the transient performance of a classical PD controller. Although the performance has not improved significantly it should be noted that the number of rules is far fewer than other supervising or hierarchical fuzzy control methods such as used by Garciabeintez et al [20], Wu *et al.* [21], Kubica [22] and Moudgal *et al.* [23], for a similar application i.e., the control of a flexible link.

The number of rules for the different groups of workers is given in table 5.5 for comparison.

These results are significant especially when considering a real-time environment such as the flexible manipulator control which requires high control update rates.

Researchers	Number of rules
Wu *et al.*	49
Kubica *et al.*	108
Garciabenitez *et al.*	464
Moudgal *et al.*	1150
Surdhar	9

Table 5.5: Summary transient performance in relation to input set quantisation

The model of the plant developed in Chapter 3 proves to be a invaluable tool for the rule analysis. The performance of fuzzy rules is usually investigated with respect to the following criteria [7]:-

Completeness of the control rules A complete rule-base refers to one where the controller can generate can generate control for any input fuzzy state (error or derivative). This implies that there must be rules for all the paths along the linguistic trajectory as described below. Completeness of rules is a more critical issue in the case of DF controllers where there is no lower level controller to fall back onto should the rules not cover the input state conditions.

Interaction between rules The interaction between rules may cause spurious output from the fuzzy controller. Such interactions or crosstalk due to the overlap of sets could occur in relational based systems where the all rules are encoded in a matrix and rule firing is based on a composition involving this matrix. Interaction is not so much of a problem for individual rule firing systems as the rules are de-coupled from each other. However, some interactions may occur in the logical construction of the controller.

Consistency of the control rules Inconsistent rules suggest diverse actions for similar input conditions. Such behaviour in the rule-base can result in unpredictable control actions. These rules may occur when the rules formulated by the operator are articulated in terms of two contradictory criteria, for example, high payload and a requirement of low torque, or when rules with the same antecedants have conflicting consequents. Inconsistent rules can be avoided through a careful rule design methodology and by studying how the rules are activated. This important point is considered in the design methodology outlined in a later section.

In order to gain more insight into the behaviour of the system, the phase-plane method, introduced by Brae and Rutherford [24] to analyse the stability of a fuzzy controller is used. This method is base on the polar plot of the error and derivative during the step change with the input space partitioned by the chosen fuzzy sets.

The linguistic trajectory or a time-history of the rule firing can easily be obtained for the step response. For the rule-base above the following closed-loop trajectory is obtained through simulation (Fig. 5.4).

PS $\rightarrow$ Z $\rightarrow$ PS $\rightarrow$ Z.

It can be seen that the rules have been designed to be consistent and that the controller is stable (The trajectory of the arm's tip is forced towards the set-point or origin and the rules that are fired behave in a stable fashion and do not push the system into an indeterminate state). Although the rulebase is complete (rules cover whole trajectory) it can be seen that all the rules are not fired implying a superfluous rule base (four rules fired out of nine). Examining the linguistic trajectory also reveals that the design philosophy of having a higher damping value just before the set point is not implemented. The damping in this case varies from PS (damping not required in the initial stages) to Z to PS (here damping should be higher) and then to Z near the set point. Use of more sets (instead of 3 for the error and derivative and output set[2]) could possibly lead to finer control in the regions of interest. The following section details a rule design methodology that leads to efficient, consistent and complete rules.

[2]this early implementation used fuzzy sets for the output

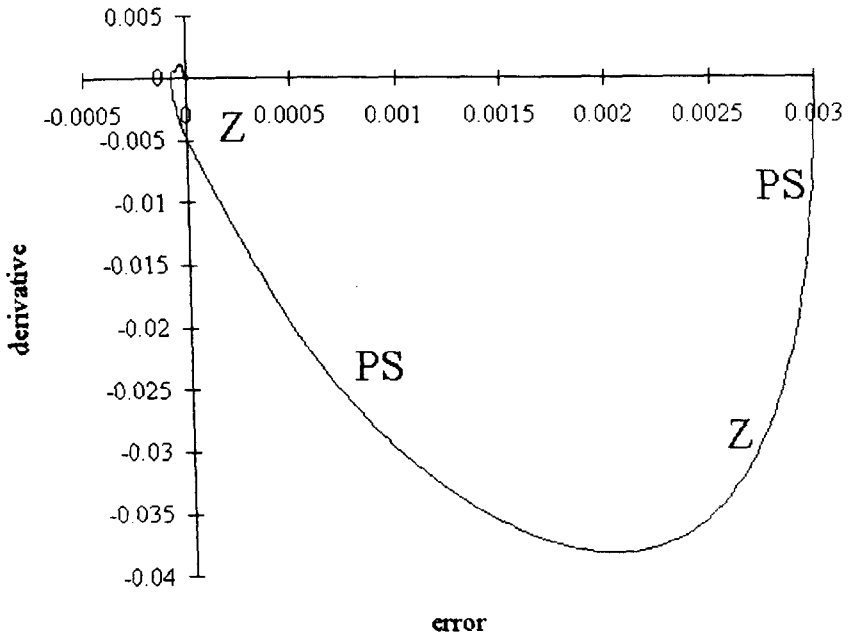

Figure 5.4: Linguistic phase plane of heuristically designed TFS based fuzzy PD controller

5.6.3 Fuzzy rule design - Modified phase plane methodology

The method of Brae and Rutherford is useful for analysis of the behaviour of the fuzzy controller giving an easily understood visual representation of which different fuzzy rules are being fired and in what sequence. They also give an insight into the stability of the closed loop system. Based on this analysis suggestions can be made for the optimisation of the rulebase, for example, unused rules may be pruned and additional safety or exception rules may be built around the linguistic trajectory. This analysis is dependent on obtaining the phase plane of a working fuzzy system either in simulation or in practice.

Rule justification through a detailed analysis of the observed behaviour and detailed annotation of such observations is both tedious and time consuming especially when the control rules need to vary dynamically. A more direct method to rule design is required.

The author investigated a *modified phase-plane methodology* for a two input variable and one output variable SF PD system. The method is not unlike the method of Brae and Rutherford in that a polar plot of the error and its derivative is required. However, there is a fundamental difference, in that instead of using the *designed* fuzzy controller, only the response under Proportional control is plotted. Also instead of the open loop response, the closed loop response was used.

With the partitioning of the input space the effect of different rules can be examined by applying rules in regions of interest. This is an important characteristic of fuzzy control which is 'localised' control as opposed to 'global' control exhibited by classical PD control. Judicious application of different derivative gains along the linguistic trajectory can enable one to tune the response of control the system.

The rule modified phase-plane design-visualisation methodology comes into its own when different payloads are applied to the system. Each different payload[3] gives rise to a different linguistic phase plane trajectory.

The simulated and measured responses are in Fig. 5.5 and 5.6 respectively. Fuzzy sets for the simulations were generated using the data processing programs in Appendix B. An input resolution of fifty was used for the fuzzy sets in the simulation. Fig. 5.5 shows also how the input space can

[3]The mass of the unloaded arm is considered to be the mass of the flexible link and the mass of the tip sensor in all the experiments.

be partitioned by fuzzy sets.

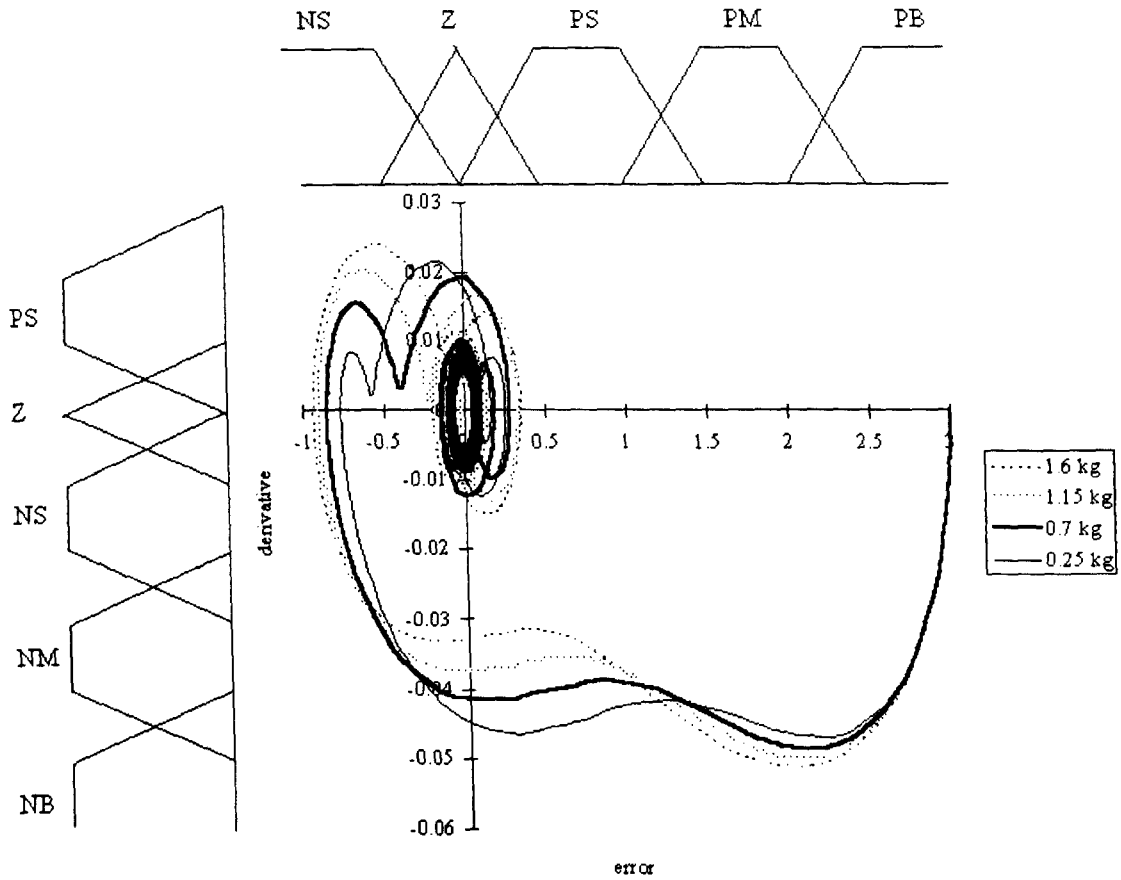

Figure 5.5: Simulated polar plots for different payloads under P control

The input space has been partitioned into 5 sets for the error and derivative. First the input space is partitioned by fuzzy sets and then rules are assigned to the respective regions of interest along the Proportional control trajectories.

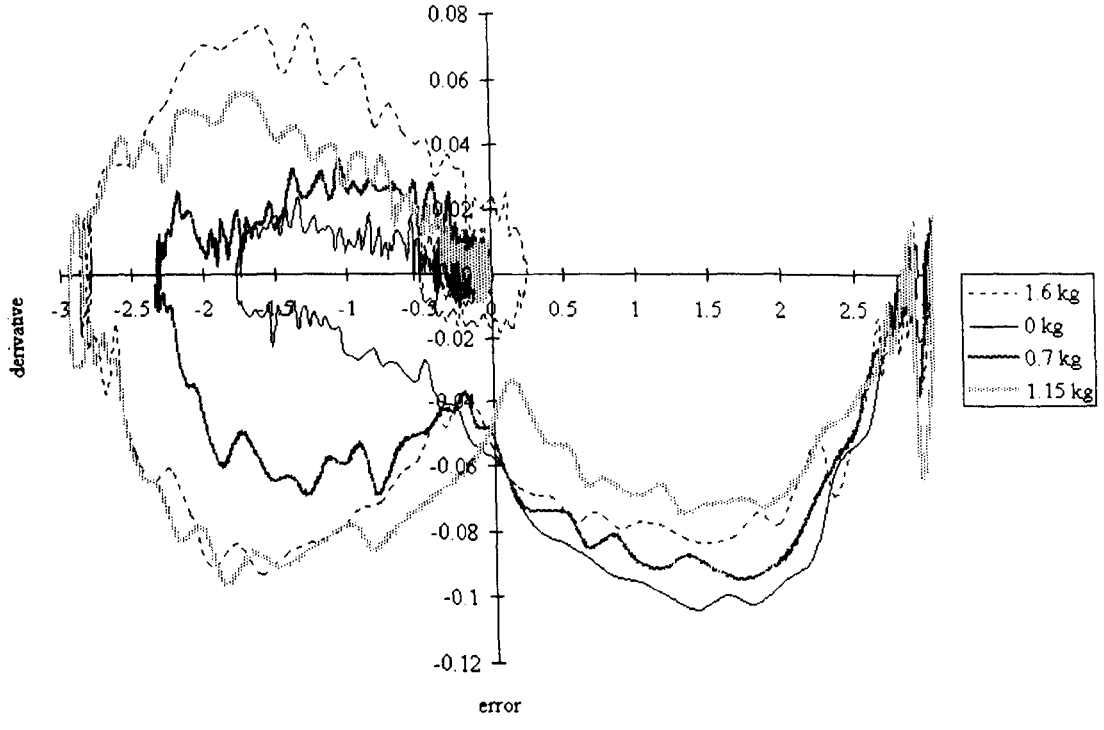

Figure 5.6: Measured polar plots for different payloads under P control.

The rules for the simulated controller are given in the matrix show in Fig. 5.7.

$\dot{e}$ \ e	NS	Z	PS	PM	PB
PS	PM	Z			
Z	Z	Z			Z
NS	PS				Z
NM		PB	PBB	Z	Z
NB			PBB	Z	Z

Figure 5.7: Fuzzy rule base (many payload design)

The distinct advantage of fuzzy control over simple look-up table or expert gain scheduling methods is seen here. The fuzzy sets cover a range of values and a rule can be applied within a particular range. Slight variations in trajectory due to a noisy signal or other unpredictable effects will not affect the control law. On the other hand look-up tables and expert systems need to have control laws coded for each eventuality of the input variables. This would require a large amount of data being stored and in may not even be possible because of the magnitude of the task of recording and coding each instance of input behaviour and a corresponding control action.

Fig. 5.8 shows the resulting simulated step response for different payloads under fuzzy control. The step response of a classical PD controller designed for critical damping of the highest payload is also shown for comparison. Table 5.6 shows the controller parameters used for each controller for the maximum payload case. It can be seen that the response of the classical controller has a much slower rise-time than that of the fuzzy controller.

Controller	K_p	K_d max	K_d steady state	T_r(s)	T_s(s)
PD	0.6	6	6	0.17	0.3
Fuzzy PD	0.6	7.75	3.5	0.08	0.18

Table 5.6: Controller parameters for maximum payload case

It should be noted that a relatively high derivative gain constant is required to give the desired damping under PD control. Studies in section 5.3.2 have shown that high derivative values are not desirable for tracking motion and result in instability when the arm is moved. High derivative in the closed loop has also been seen to make the system very sensitive to noise. In contrast the fuzzy controller injects high derivative a little before the system reaches set-point and then sets it to a low 'safe' steady state level (Fig. 5.9).

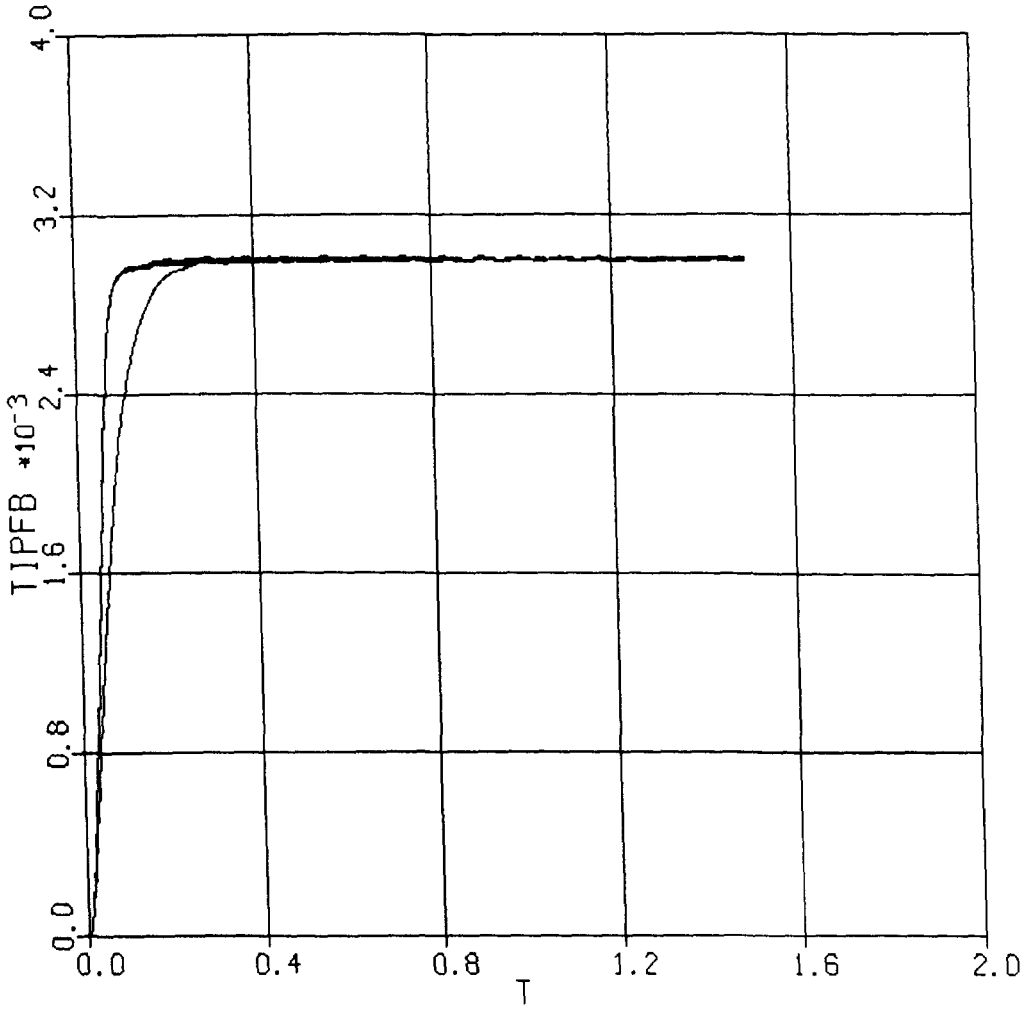

Figure 5.8: Step responses with different payloads under fuzzy PD control. The responses for the different payload cases (0, 0.7, 1.15, and 1.6 kgs) under fuzzy PD control are indistinguishable. In contrast, the classical PD response, designed for critical damping (1.6 kg), is seen to have a slower rise and settling time.

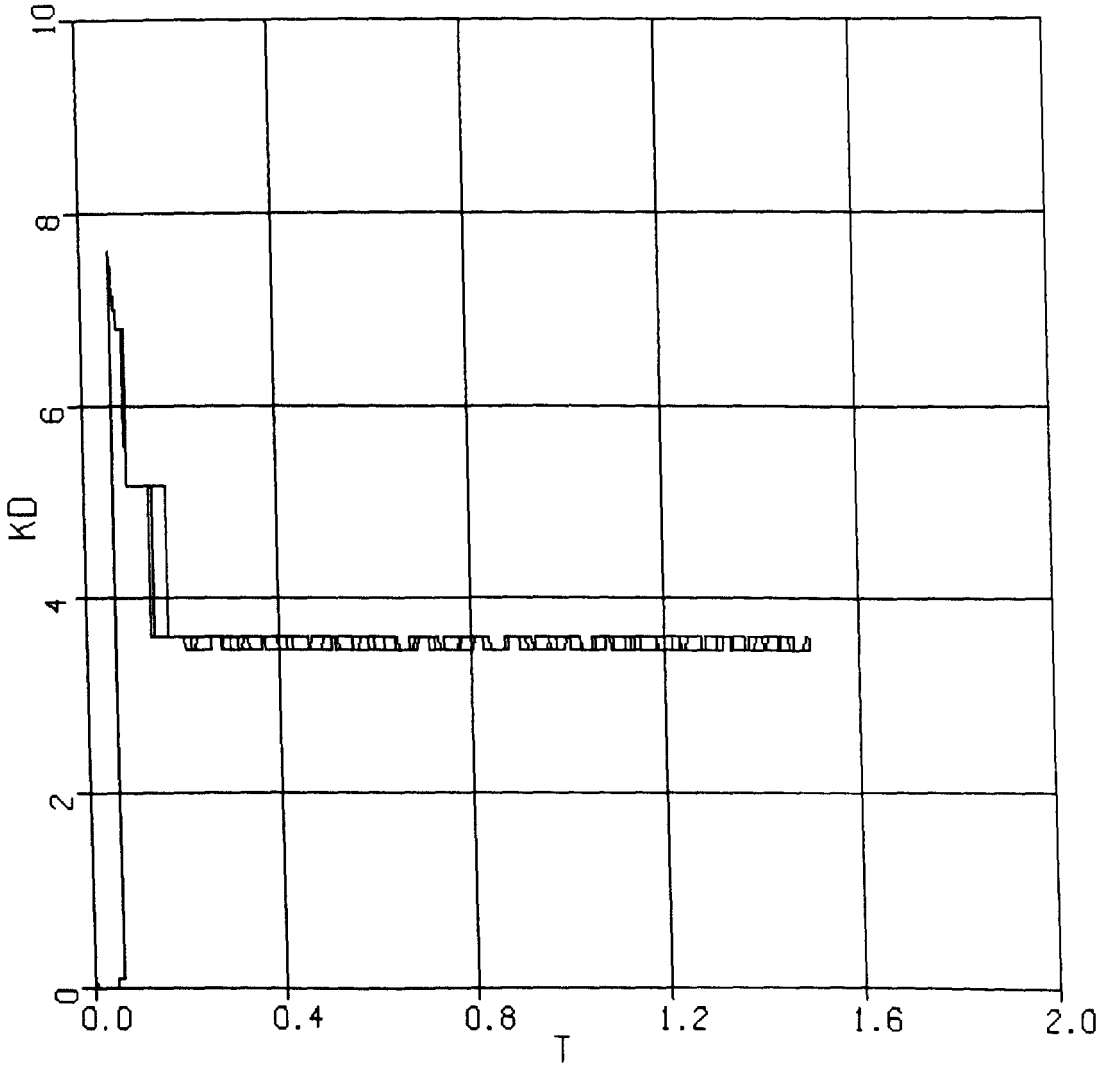

Figure 5.9: K_d response for different payloads under fuzzy PD control.

Rule design - discussion

From these observations it can be seen that the effect of K_d is analogous to that of the strength of braking for a moving vehicle. A positive high value applied just before the destination should bring the vehicle to an abrupt stop. It is important to note that these experiments have been easily facilitated by use of a model of the system which is a safe mode of investigation before the experiments are carried out in practice.

We can see that the approach for design of fuzzy rules via the modified phase plane technique is exactly the same for no payload and with different payloads. Rules implement the simple control philosophy, analogous to stopping a speeding car of breaking more heavily just before the destination set-point. The simulations show that fuzzy PD control yields optimal transient responses in comparison to classical PD control, where for the many payloads a 'worst-case' design is required. A further advantage of tuning the K_d gain constant is that it can be switched to a low safe value that is desirable for tracking. The classical 'worse-case' design on the other hand leaves K_d at an undesirably high value making the system less robust to noise and disturbances.

There are differences in the polar plots of the simulated and measured responses. The main difference is that the measured response is more under-damped than the simulation. This high overshoot can be attributed to the effects of friction which require a greater torque to be generated initially in the actuators. Once the stiction has been overcome there is a whiplash effect causing the higher than expected overshoots. The region where the high torque is generated can be identified as a small glitch in the early part of the response. The derivative measurements tend to be higher than the simulation as a result of this effect. The implications on the design of the fuzzy controller are that the fuzzy sets must be correctly scaled to cover the entire universe of discourse. This is easily done by extending the range of the derivative sets by a factor of two. For example the range of Z (derivative) is extended from $\pm$ 0.01 to $\pm$ 0.02 and NS (derivative) is extended from 0 - 0.03 to 0 - 0.06. This scaling still satisfies the criteria for the fuzzy sets as outlined in section 4.4.

The discrepancy between simulation and measurement would lead to different rulebases being designed. However, this is not a major let down, as rules can be still be designed for the practical system, implementing the same control philosophy and capture of artisan knowledge. This is an advantage of using the supervisor based and modified linguistic phase-plane approach over a conventional model based approach.

5.6.4 Adaptive fuzzy control

Self organising fuzzy control was briefly mentioned in an earlier chapter as a logical progression from the heuristic design process. The aim of self organisation is to improve on the performance of the fuzzy controller through adjusting the fuzzy systems parameters. Candidates for adjustment are the variable membership functions, the input and output scaling and the rules themselves. The former two parameters remain fixed in the design of the TFS based fuzzy controller in order to conform with the design criteria laid down in the previous chapter. Adjustment of the rules in a bid to improve performance was investigated briefly.

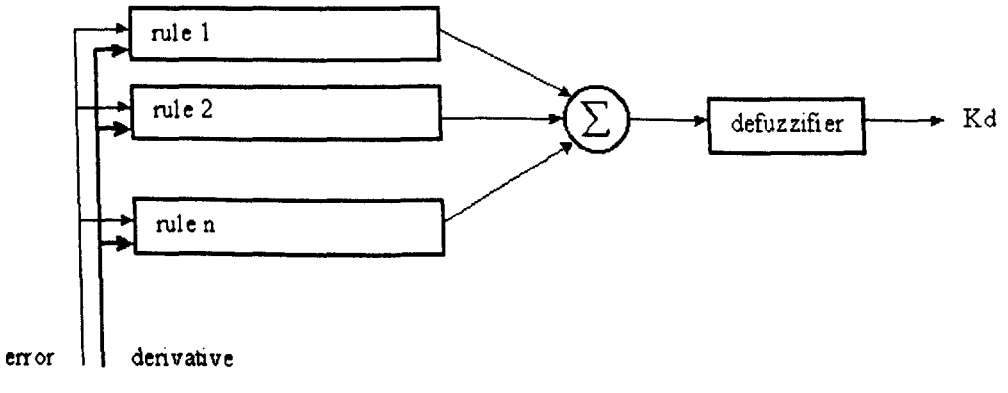

Figure 5.10: Similarity between fuzzy control and the CMAC neural network

The fuzzy algorithm can be made adaptive by adjusting the degrees of confidence for the individual

rules according to the performance of the plant. This is done by making the weights associated with each rule variable. The Least Means Squared (LMS) algorithm was used to update the rules locally in a similar manner to Albus's cerebral articulation model controller (CMAC) neural network [25]. The LMS adaptation law is given in Eq. 5.2 for the k^{th} rule:-

$$w_{k_{t+1}} = w_{k_t} + 2.\alpha.error_{k_t}.\mu_{k_t} \tag{5.2}$$

Here $w_{k_{t+1}}$ and α are the adapted rule confidence for the k^{th} rule and learning coefficient respectively. Fig. 5.11 shows the corresponding simulated step response. For comparison the response for a non-adaptive version designed with the proposed methodology is also given.

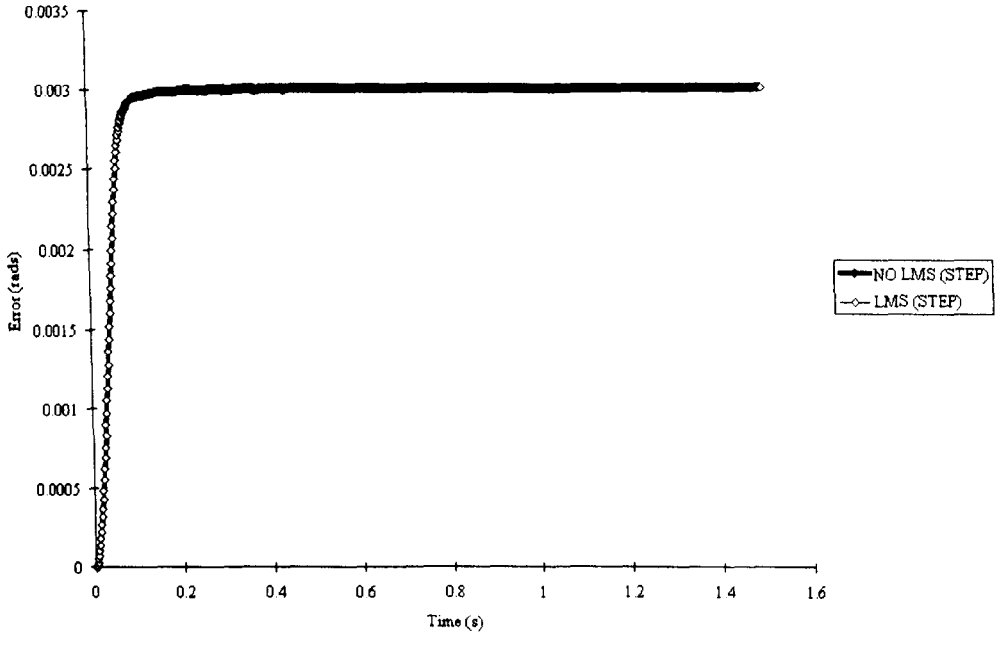

Figure 5.11: Comparison of adaptive and non-adaptive fuzzy PD control

The LMS algorithm does not seem to make a difference to the transient performance in this case. It is possible that the learning parameters were not chosen correctly (for example the value of the learning coefficient, α). Another possibility is that the rules have already been adequately designed and hence do not require further adaptation. The variation of the consequent is shown in Fig. 5.12 for the non-adaptive and adaptive fuzzy controllers.

The K_d value is initially similar in both cases but just before 0.2 s the adaptive K_d diverges from the non-adapted version. The final value of K_d for the adaptive LMS case is near the maximum allowable K_d value. Although the transient performance is similar in both cases the adaptive value leaves K_d in an undesirably high state where the system is more likely to be unstable and is more sensitive to noise.

However, there is an important observation to be seen from this work. It shows that in the critical region, between 0 and 0.18 s, that the rules defined by the modified phase-plane technique do not have to be adapted and are well designed according to the LMS $error^2$ index.

5.6.5 Defuzzification and denormalisation

Defuzzification is the final stage in the fuzzy inference stage. In this stage output fuzzy sets are converted to crisp single values. The previous chapter evaluated different fuzzy defuzzification methods and determined that weighted average defuzzifiaction was the most suitable for real-time applications where the outputs consequents were composed on singleton values. This type of defuzzification was also attractive because it lends itself easily to adaptive fuzzy control as examined in the previous section.

Usually the denormalisation stage is required to convert crisp values computed as a result of

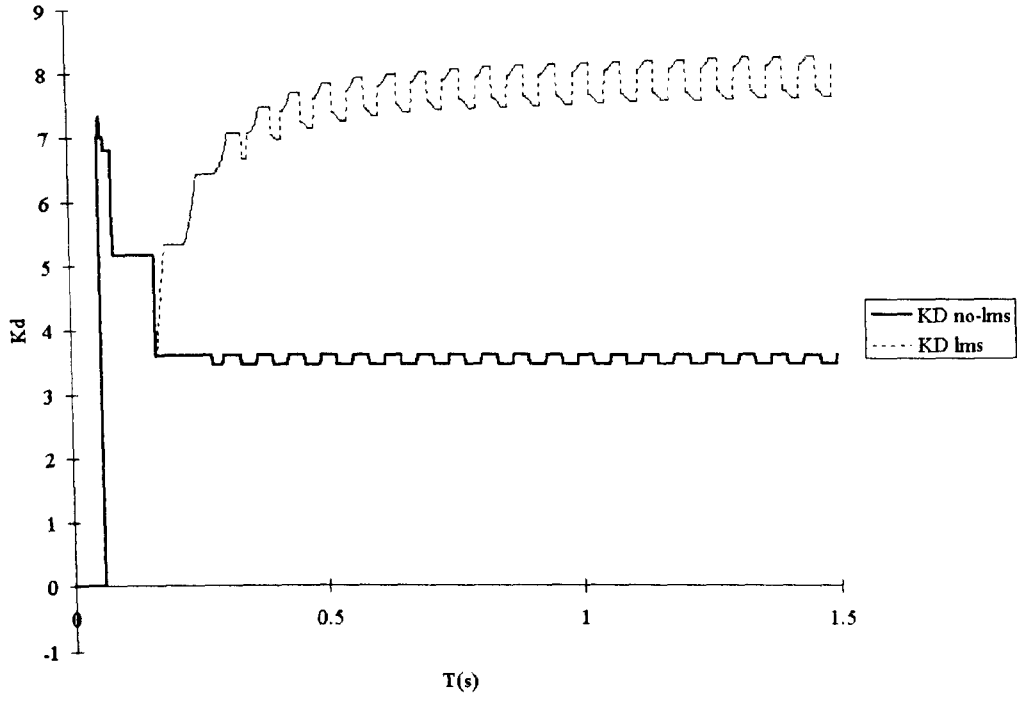

Figure 5.12: Change in the confidence parameter

defuzzification into the required output range. This can be carried out by a simple uniform scaling, the inverse of Eq. 5.1. This calculation need not be carried out if the computed crisp values are in the required range of the output. i.e., if the output singleton values are not normalised. The fuzzy implementation described tin this thesis did not require the additional de-normalisation calculation.

5.7 Design summary and discussion

This chapter has covered in detail the implementation of a real-time fuzzy PD controller applied to the TFS based long reach robot.

An important stage in the design process is the choice of variables for the double antecedant and single consequent fuzzy controller. The antecedants comprise of the position error and its rate. The consequent, or fuzzy control output variable is the derivative gain constant K_d. The choice of the antecedants has been made based on the fact that they are readily available from the TFS based sensor arrangement. The consequent was chosen on the basis that it exerts the most influence in the damping of tip vibrations.

Rule design has been carried out using a modified phase plane methodology. This methodology proved to be more useful for the design of fuzzy rules than Rutherford and Brae's method which is probably better as a post implementation analysis tool. The design methodology leads to complete, and consistent control rules and can cope with parameter variation and uncertainty due to different payloads. The design methodology is especially suited for the design of supervised systems where parameters are to be adjusted by the jacketing fuzzy controller. It should be noted that a model of the system is *not* required using this design methodology. This can be seen by the fact that it is possible to design rules for the practical system based on simple closed loop measurements and P control.

Adaptive fuzzy control has been examined briefly by adjusting the confidence of each fired rule according to the LMS algorithm as in the CMAC neural network. A step response that is nearly identical to the heuristically designed version is obtained using the adaptive scheme. A drawback of LMS adaptation is that leaves the K_d parameter undesirably high at the end of the step response. This would increase the controller's sensitivity and reduce its relative stability. These experiments also verify that the heuristically designed rules minimise the error squared criterion in the critical

section of the transient response.

References

[1] Astrom K.J and Hagglund T. PID control. In Levine W.S., editor, *The control handbook*, pages 198–209. CRC Press, IEEE press, 1996, 1996.

[2] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[3] Zeigler J.G. and Nichols N.B. Optimum settings for automatic controllers. *Trans ASME*, 64:759–768, 1942.

[4] K.L Chien, Hrones J. A., and Reswick J.B. On the automatic control of generalised passive systems. *Trans ASME*, 74:175, 1952.

[5] Cohen G.H. and Coon G.A. Theoretical considerations of retarded control. *Trans ASME*, 75:827, 1953.

[6] Lee C.C. Fuzzy logic in control systems: Fuzzy logic controller - Part i. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):405–417, 1990.

[7] Pedrycz W. *Fuzzy control and fuzzy systems.* Research Studies Press Ltd., 1993.

[8] Surmann H. and Ungering A. P. Fuzzy rule based systems on general purpose processors. *IEEE Micro*, pages 40–48, 1995.

[9] Kosko B. *Neural Networks and Fuzzy Systems.* Prentice Hall, UK, 1992.

[10] Yi S. Y. and Chung M. J. Robustness of fuzzy logic control for an uncertain dynamic system. *IEEE Transactions on Fuzzy Systems*, 6(2):216–225, 1998.

[11] Maiers J. and Sherif Y.S. Applications of fuzzy set theory. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):175–189, 1985.

[12] Yao S., Wei C., and He Z. Evolving fuzzy neural networks for extracting rules. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 361–367, New Orleans, Louisiana, USA, 1996.

[13] Nelles O., Fischer M., and Muller B. Fuzzy rule extraction by a genetic algorithm and constrained non-linear optimisation of membership functions. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 213–219, New Orleans, Louisiana, USA, 1996.

[14] Ghwanmeh S.H., Jones K.O., and Williams D. On -line performance evaluation of a self-learning fuzzy logic controller applied to a non-linear process. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 394–400, New Orleans, Louisiana, USA, 1996.

[15] Yurkovich S. A control engineers perspective on fuzzy control. In Bastin G. and Gevers M., editors, *Plenary Lectures and Mini-Courses of the European Control Conference, ECC97*, pages 75–105. European Control Conference, 1997.

[16] Surdhar J. S. and Korhonen J. A fuzzy PD controller applied to a long reach manipulator. In *Middlesex University Faculty of Technology Technical Report Series ,MUCORT - Middlesex University Research Conference*, Middlesex University, London, 1995. ISSN 1362-2285.

[17] Editor. User manual for PC-30B/C/D. Technical report, Amplicon Liveline Limited, Centenary Industrial Estate, Brighton, East Sussex BN2 4AW, 1990.

[18] Golten J. and Verwer A. *Control system design and simulation*, chapter 9, page 323. McGraw Hill Book Company Europe, 1991.

[19] Surdhar J. S., White A. S., Gill R., and Mistry G. An efficient fuzzy algorithm applied to a flexible link 1dof robot. In *Proc. of the FUZZ-IEEE '96 International Conference on Fuzzy Systems*, pages 68–72, New Orleans, Louisiana, USA, 1996.

[20] Garciabenitez E., Yurkovich S., and Passino K. Rule-Based supervisory control of a two link flexible manipulator. *Journal of Intelligent and Robotic Systems*, 7:195–213, 1993.

[21] Wu J. and Tsuei Y. G. Comparison of fuzzy logic and self-tuning adaptive control of single-link flexible arm. *Mechatronics*, 3(4):451–464, 1993.

[22] Kubica E. and Wang D. Fuzzy control strategy for flexible a single link robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 236–241, 1993.

[23] Moudgal V. G., Kwong W. A., Passino K. A., and Yurkovich S. Fuzzy learning control for a flexible link robot. *IEEE Transactions on Fuzzy Systems*, 3(2):199–210, 1995.

[24] Brae M. and Rutherford D.A. Theoretical and linguistic aspects of the fuzzy logic controller. *Automatica*, 15:15–30, 1979.

[25] Albus J.S. A new approach to manipulator control: The cerebral model articulation controller (cmac). *J. Dynam. Sys. Measurement Control*, 58:220–227, 1975.

Chapter 6

Controller implementation and performance analysis

6.1 Introduction

Real-time implementation of the fuzzy PD controller on distributed parallel processors is the final stage in the controller development for this research programme. Issues such as partitioning the fuzzy controller algorithm into separate concurrent tasks, input and output filtering and task scheduling are addressed.

The performance of the fuzzy controller was measured and compared with the simulation results. The performance is investigated from both a control viewpoint, where control performance is evaluated using standard transient and tracking responses and robustness testing, and a real-time processing viewpoint, where performance of the controller is evaluated in terms of processing speed and memory requirements.

6.2 Transputer implementation

The initial proof of concept fuzzy PD controller was implemented sequentially on PC with a INTEL 386 processor running at 30 MHz. Responses using control sample rates of 10 ms for a single axis were obtained [1]. Although this sampling rate is suitable for controlling a plant whose frequency of interest is under 20 Hz, the processing time would double for control of a second axis. Addition of more rules would further slow down the overall computing time.

Parallel processing using *transputers* was chosen as a more flexible development platform capable of delivering high bandwidth and enough room for experimentation with greater numbers of rules. This choice was based on the premise that fuzzy controllers are inherently parallel [2]. It was envisaged that a substantial increase in processing speeds could be attained if the fuzzy rules were processed concurrently. Also the control of each axis could be done on separate processors again reducing the processing time and at the same time de-coupling the control for each axis. The transputer based approach provided a flexible development platform where the notion of parallel fuzzy processing could be investigated and the number of rules varied in a systematic manner.

A transputer may be described as a multiple-instruction multiple-data (MIMD) machine where multiple sets of possibly different instructions are executed concurrently on multiple data sets [3]. The parallel processing ability of the transputer is derived from a novel computer architecture developed by Inmos Ltd. as a microprocessor. This processor includes not only processor and memory components, but also a channel, or 'link', for communicating with other transputers and to other devices. This link and its properties comprise one of the fundamental distinguishing features of this architecture. Communication across the link takes place only once both ends are ready, so that events are synchronised [4].

The multi-processing ability of the transputer is used to reduce the overall processing time of the

control algorithm. This is achieved by partitioning the control algorithm, using different 'processors' to carry out calculations concurrently. A further advantage of using this distributed processing system is that the extension of control to multiple axes can be easily achieved by simply adding on more processors. A critical factor, however, is that the inter-processor communications through the physical links between the processor tasks has to be tightly monitored to avoid bottlenecks. The T805 datasheet [5] specifies a link clock period as $\approx$ 50 ns (for a link speed of 20 Mbits/s).

6.3 Parallel implementation

As already indicated, the fuzzy algorithm is inherently parallel [2] and can be visualised as shown in Fig 5.10. Using transputers this can be achieved by multi-tasking, processing each rule as a separate thread or setting up separate processes tasks for the rules. This section investigates transputer based implementations of the fuzzy controller where the processing time of the rules is reduced by concurrent processing.

6.3.1 T800 implementation - single axis

The parallel implementation of the fuzzy algorithm on a single T800 transputer is shown below. The processing of the fuzzy rules was designed to be carried out on three separate slave tasks (each task processing an equal number of rules in a semi-parallel fashion). This design was adopted with flexibility in mind, retaining a four-way connection between tasks, should the tasks themselves later be distributed onto separate processors[1]. Executing each of the rules on the separate slave tasks can make the processing of the rules completely concurrent.

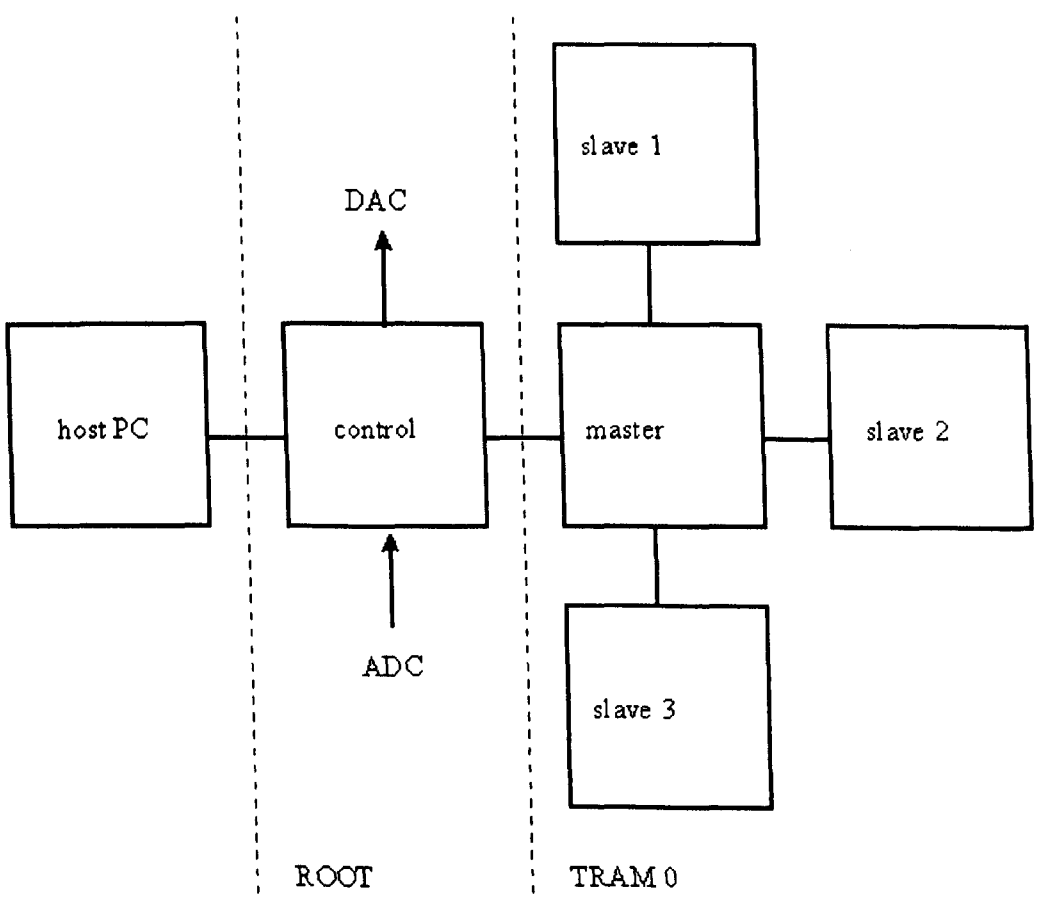

Figure 6.1: Parallel implementation of fuzzy rules (single axis)

A direct comparison of the PC and transputer implementation speeds was carried out for the proof of concept controller described in section 5.6.1. N.B. This controller had output set as opposed to output singletons. The transputer implementation yielded a control sampling rate of 2 ms in

[1] processors have only four bi-directional connections or links for inter-connectivity

comparison to the 10 ms obtained by the PC implementation. A later upgrade from the T800 transputer (20 MHz) to the T805 (30 MHz) gave a sample time of 1.2 ms for a single axis for the same system.

It should be noted that the increased sampling rate is not wholly due to the transputer implementation as the data acquisition card was also changed[2]. These observations indicate that concurrent processing is of benefit especially in the context of obtaining faster controller sample rates.

Fig. 6.2 shows the transient response obtained by implementing the proof of concept controller on the T800 transputer. N.B. for these results the fuzzy rules were adjusted so as to be conformant with the design philosophy of having a high derivative action in the critical region of the response. This adjustment is based on the phase-plane analysis of section 5.6.2. The rule matrix is depicted in Fig. 6.3.

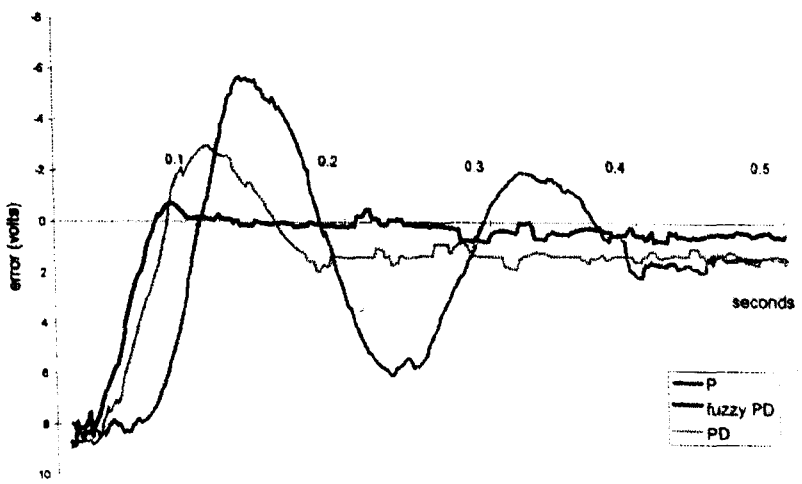

Figure 6.2: Step response for fuzzy PD controller (no payload)

$\dot{e}$ \ e	PSE	PME	PBE
NSE	Z	Z	Z
NME	PS	PS	PS
NBE	PM	PM	PM

Figure 6.3: Heuristically designed fuzzy rule-base after analysis

The response shows a marked improvement in overshoot over the heuristically designed fuzzy controller.

[2]The PC based PC30B gave a data sampling rate of 30 kHz as opposed to a data sampling rate of 40 kHz obtained via the custom made university of Bangor cards

6.3.2 T805 implementation - dual axis

The implementation of Fig. 6.1 was suitable for single axis control. For dual axis control, the hardware was upgraded by adding on two more processors. The dual axis architecture is shown below.

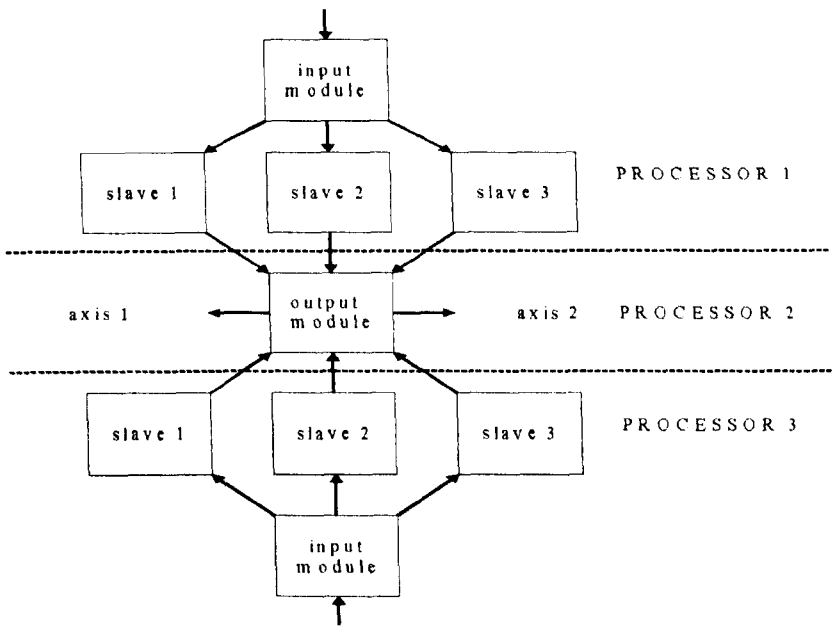

Figure 6.4: Parallel implementation of fuzzy rules (dual axis)

The new processors (T805) added were a later version to the existing T800 processor, the main difference being that the more recent version has an extended instruction set to support debugging. This family of transputers is suitable for the application being considered as they have been designed for numerical applications with 32 bit or 64 bit floating point support. The T805 had a slightly faster clock speed of 30 Hz as opposed to 25 MHz. The processors were mounted on a TMB08 motherboard with 1MB external RAM. Both types of processor have 4KB of internal fast RAM. ADC and DAC cards with high sampling rates (18 and 16 microseconds respectively) were used for data acquisition.

Datasheets for the motherboard, processors and the DAC and ADC cards are given in Appendix D. The configuration file for setting up the processes and their inter-connection is given in the file *dual.cfg* in Appendix C . The external links to the transputers between the ADC and DAC were configured via the C400 programmable link switch [3].

Before implementation an assessment of the processing time for the different functions in the fuzzy PD controller was made. The main processing stages of the fuzzy controller are identified as:-

1. Inference,
2. defuzzification,
3. and output processing.

Table 6.1 below gives the relative processing times of for each stage of the fuzzy PD controller with 14 rules. These measurements were made on a T805 transputer and averaged out over 1,000 loops. All the processes were run sequentially on a single T805 transputer. The fuzzy architecture comprised of a 10 fuzzy sets for the antecedants (5 for error and 5 for derivative) and 5 output singletons. *Conjunction* of the input sets was carried out using the *min* t-norm and rule firing was carried out by the product or *scaled inference*. The aggregation of fired outputs was carried out by taking the sum of the fired elements and weighted average defuzzification was used to produce the crisp output.

Based on the observations above the inference stage is seen to take the longest processing time. This observation is consistent with those of 32-bit implementations by Surmann and Ungering [6].

Function	Execution time / (μs)
inference	25
defuzzification (5 singletons)	4.5
output processing	3.3

Table 6.1: Relative processing times for serial version of fuzzy PD controller

Table 6.2 gives the relative processing times of for each stage of the fuzzy PD controller when the rules are implemented in parallel. It can be seen that the execution times of the various stages of the algorithm are carried out extremely quickly (order of tens of microseconds). These speeds are similar to those reported by Costa *et al.*[7] for dedicated fuzzy processor implementations. Surmann and Ungering also achieve similar speeds for similar sized fuzzy systems coded in assembler.

Function	Execution time / (μs)
inference	13
defuzzification (5 singletons)	4.5
output processing	3.3

Table 6.2: Relative processing times for parallel version of fuzzy PD controller

The speed-up obtained through parallel execution of the rules (inference) is determined below:-

$$Speedup = \frac{Sequential\ processing\ time}{Parallel\ processing\ time} = \frac{25}{13} = 1.92 \tag{6.1}$$

It should be noted that this implementation did not process each of the rules as separate threads on the slave tasks. Instead, processing was carried out in groups of rules. The 14 rules were divided as follows:-

Task	Number of rules
Task 1	5 rules
Task 2	5 rules
Task 3	4 rules

Table 6.3: Distribution of rules on tasks

Faster rule execution speeds could theoretically be achieved if each of the rules were to be processed in parallel. Faster processing however, was unnecessary. The reason for this was that achieved controller sample time of 237 μs (parallel) for the algorithm was already more than suitable for control of a system with a nyquist frequency of just under 20 Hz (cf. Chapter 3).

There is a disparity in the timings i.e., a total processing time of just over 25 μs versus the obtained control sample rate of 237 μs. One would not expect such a difference especially since the ADC and DAC are capable of 18 and 16 μs sample rates respectively. This disparity could be due to the following reasons:-

1. Calculation of the signal derivative is carried out in the external I/O interrupt. Smaller interrupt latency can be achieved if less processing is carried out from within it. A method of buffering the incoming stream of data and processing it outside the interrupt is more desirable.

2. The C400 programmable link switch causes a 15-20% reduction in I/O link speed [3].

The parallel implementation of the fuzzy PD controller has been shown to reduce processing times for the algorithm. As it stands the algorithm is already very 'thin' in terms of processor overhead. This is because processing speed was taken into consideration from the outset when designing the fuzzy system. Antecedants implemented as a look-up table, scaled inference and weighted average

defuzzification of the singleton outputs resulted in a quick algorithm. Coding the controller in a low level assembler language can potentially further improve processing speed of the algorithm.

Complete parallel implementation of the algorithm via processing each rule as a thread on the slave tasks has the potential of making the algorithm even faster. This would be of great benefit where there are a larger number rules to be processed and for systems where bandwidth constraint is more stringent. However, before gains are made through complete parallel processing of the rules, the bottleneck due to I/O data sample rate processing needs to be addressed.

6.4 Input resolution

Although the number of sets covering the Universe of Discourse (UoD) influences the quality of the controller output, it is more important to consider the input resolution of the individual sets as this affects the accuracy, overshoot and damping effects of the fuzzy control system [8]. For a linear dynamic system Boverie *et al.* found that fine quantisation or a high input resolution was found to generate an overshoot whereas coarse quantisation favoured damping.

In these experiments only the input resolution of the of the input sets was varied. All the other parameters of the controller remained the same viz., set shapes, rules and defuzzification method. Rules for the fuzzy controller were designed using the phase plane methodology of rule formulation for different payloads described in section 5.6.3.

For data processing, a simple program written in C++ was used to generate the quantised fuzzy sets. The source code is given in Appendix B. The simulation experiments were carried out on an unloaded arm. The sampling rate of the simulation was adjusted to twice that of the sampling frequency of the practical system by adjusting the ACSL integration step size ($MAXT = 0.0001$ s). Fig. 6.5 shows the step responses obtained with different input resolutions of the antecedant sets.

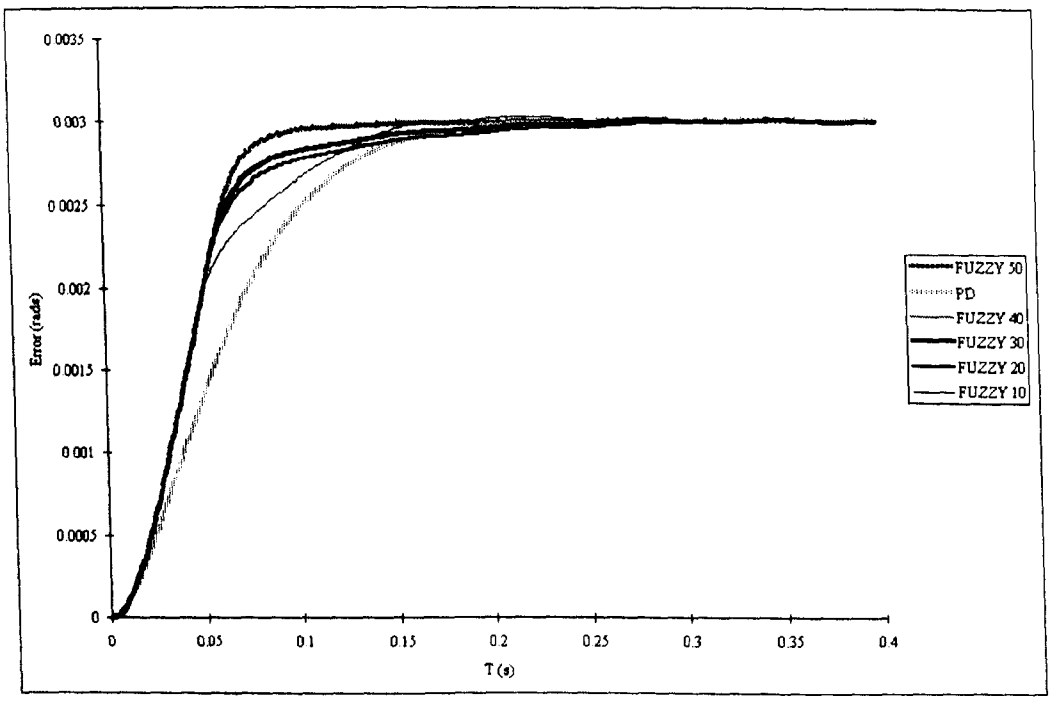

Figure 6.5: Simulated step response under fuzzy control with varying quantisation of the input fuzzy sets

Table 6.4 summarises the above observations.

It can be seen that the higher quantisations yield better transient response performance. $N_q = 50$ and $N_q = 40$ give the fastest rise-times and settling times in comparison to the other levels and PD control. The fuzzy controller in these cases gives a 33% faster rise-time and the settling time is 55% faster. The fuzzy controller transient performance deteriorates as N_q is reduced. For $N_q =$ 30, 20 and 10 fuzzy controller rise-time is slower than the classical PD rise-time by up to 25% and

N_q	t_s (s)	t_r (s)
50	0.16	0.05
40	0.16	0.05
30	0.31	0.25
20	0.32	0.29
10	0.32	0.3
PD	0.24	0.11

Table 6.4: Summary transient performance in relation to input set quantisation

the settling time slower by 72%.

These observations show that it is crucial to choose the right N_q for the fuzzy controller. Analysis of the variation of the of the derivative gain constant gives more insight into the behaviour of the fuzzy controllers with different quantisation levels. Fig.6.6 depicts the K_d behaviour in response to an input step.

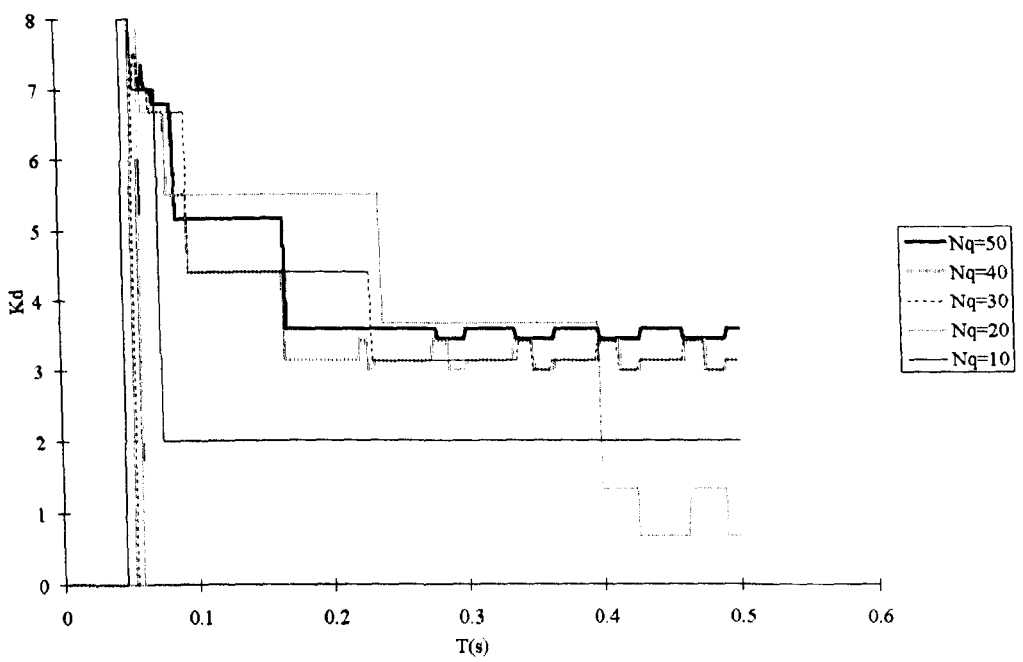

Figure 6.6: Simulated variation of the consequent K_d with resolution of the fuzzy sets

The responses for $N_q = 50$ and 40 are very similar in that they rise quickly to the maximum value just after 0.05 s and then make a more gradual descent to the lower and safer values of $K_d = 3.3$ and 3 respectively.

In contrast $N_q = 10$ behaves more like a short spike of damping rising quickly to maximum value at 0.04 s and then falling sharply to $N_q = 20$ at 0.075 s. The effect of the spiked damping is evident in the transient response. Until 0.05 s the response has a quick rise-time. Between 0.05 and 0.075 s when the damping is quickly removed, the plant response slows considerably. After 0.075 s, K_d is constant and too low ($K_d = 2$) to damp any overshoot caused by the initial stage with no damping.

Lower input set resolutions are desirable for real-time implementation in terms of memory storage overhead. An input set of $N_q = 40$ would require 40 bytes of storage. In contrast the 8-bit general μprocessor implementation of Surmann and Ungering [6] used 256 bytes per set. This represents a memory saving of 84% per set relative to [6].

From a real-time perspective the implementation of fuzzy sets as arrays proves to be much quicker than computation of the membership functions online. A pre-computed array implementation requires a simple de-reference versus a minimum of 4 mathematical operations (2 subtractions and 2 multiplications) and 4 comparisons[9].

Source code for the efficient (in terms of both memory and speed) parallel implementation of the fuzzy PD controller is given in Appendix C.

6.5 Sampling and filtering issues

Practical control systems are subject to noisy signals, and when sampling of data is involved, a phenomenon known as aliasing. Corruption of the measured signal by noisy signal measurements or aliasing can cause poor performance in the controller. It is important to ensure that the signals that the controller is dealing with correct representations of what are being measured from the system. This section outlines the steps taken in order to ensure faithful reproduction of the measured signals for controller.

A data acquisition program written in Labview (Appendix F) was used to analyse the unfiltered data being measured from the tip sensor. A sampling rate of 0.2375 ms (4220 Hz) was used to collect the data. This sample rate corresponds to that obtained for the parallel implementation of the dual axis controller described in section 6.3.2.

The measured signal was extremely noisy. The noise was due to the use of PWM amplifiers that were used to drive the manipulator actuators [10]. Sampling at a rate of 4220 Hz resulted in a very noisy first order derivative and it was clear that this derivative could not be used in any kind of control where derivative action was required. Reducing the rate at which the derivative was calculated to 52 Hz gave a less noisy response. This sample rate still complies with Shannon's sampling theorem requiring a sample rate of at least twice the frequency of interest (in our case the Nyquist frequency is $\approx$ 20 Hz).

Reducing the derivative sampling rate solved the problem of noisy data but was not the ideal solution. The control system would still be susceptible to aliasing. Aliasing occurs in sampled data systems when higher frequency harmonics of the signal are mistaken for the signal being measured at a lower frequency. In order to prevent these effects it is necessary to use an anti-aliasing filter [11]. An analogue elliptic filter (by Kemo Ltd.) with a cut-off at 20 Hz with a 24 dB per octave rejection was used. The cut-off frequency was set by resistor dial. The anti-aliasing filter is also important for the prevention of the effects of observer spillover described in Chapter 3.

6.6 Performance analysis - Controller

6.6.1 Transient response

This section shows the results obtained from implementation of the fuzzy PD controller on the different processors viz., the T800 and T805. The former implementation considers the no payload case .

Different payloads

For the many payload design rule derivation was carried out using the measured phase plane response of Fig. 5.6. It should be noted that the derivative sets were re-scaled in order to fully cover the entire universe of discourse c.f. section 5.6.3. The ranges for the error sets remained unchanged from the simulation. The derived rules are given in the matrix shown in Fig. 6.7. Values of the consequent (K_d) are also given in the matrix as underscores.

It should be noted also that the damping response is slightly different from the simulation in that the damping is not injected suddenly into the system. This is because the practical system exhibits a non-linearity due to stiction in the beginning of the response (manifested by the little 'bump' in at the early part of the response). The result of this is to cause a whiplash-like effect in the response of the arm. The way to compensate for this is to incorporate damping at a much earlier stage in the response rather than trying suddenly damp the whiplash at a later stage. This would require extremely high derivative gains and is detrimental to the transient response (cf. Section

6.6.2).

$\dot{e}$ \ e	NS	Z	PS	PM	PB
PS	PS_4	PS_4	PS_4		
Z	PS_4	Z_0			PS_4
NS	PM_6	PS_4			PM_6
NM	PB_8		PM_6	PB_8	
NB					

Figure 6.7: Fuzzy rule base (many payload design practical implementation)

The resulting step responses for the different payloads in the right hand side direction are shown in Fig. 6.8.

The transient response for the PD controller is also shown in Fig. 6.9 for comparison. It should be noted that transient responses are for controllers that were stable during tracking as well.

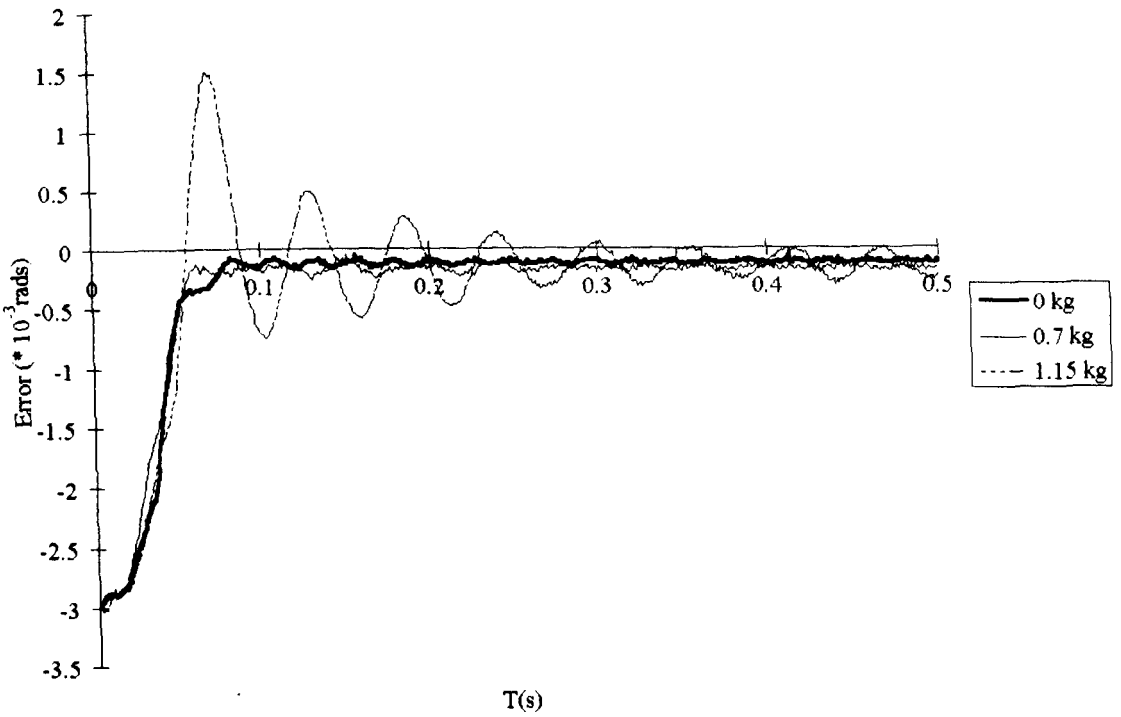

Figure 6.8: Step response for the fuzzy PD controller (different payloads)

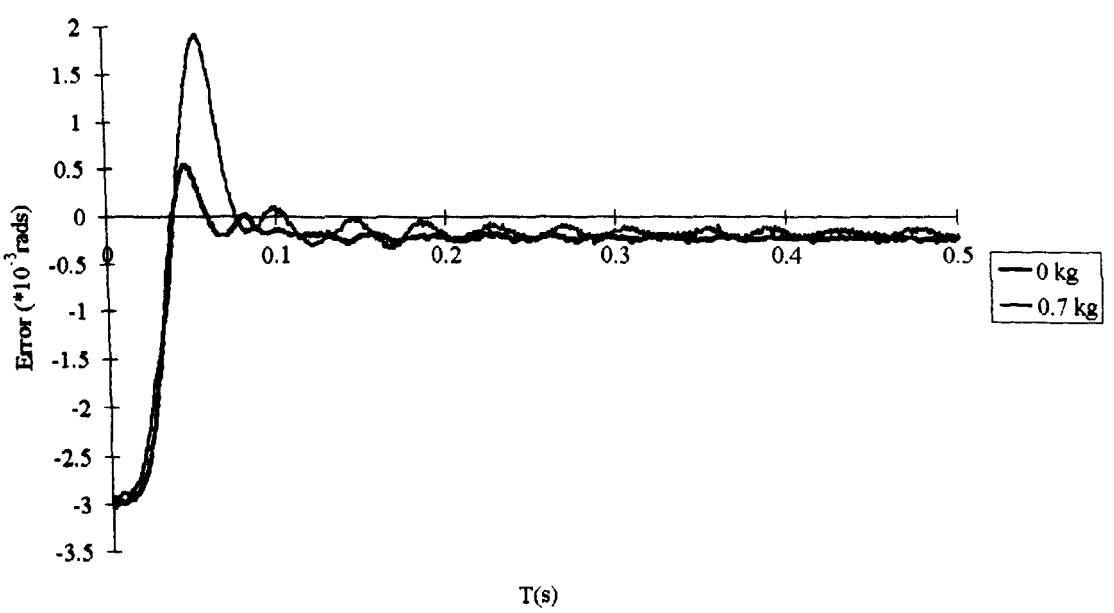

Figure 6.9: Step response for the PD controller (different payloads)

The linguistic trajectory for the design rules in Fig. 6.10.

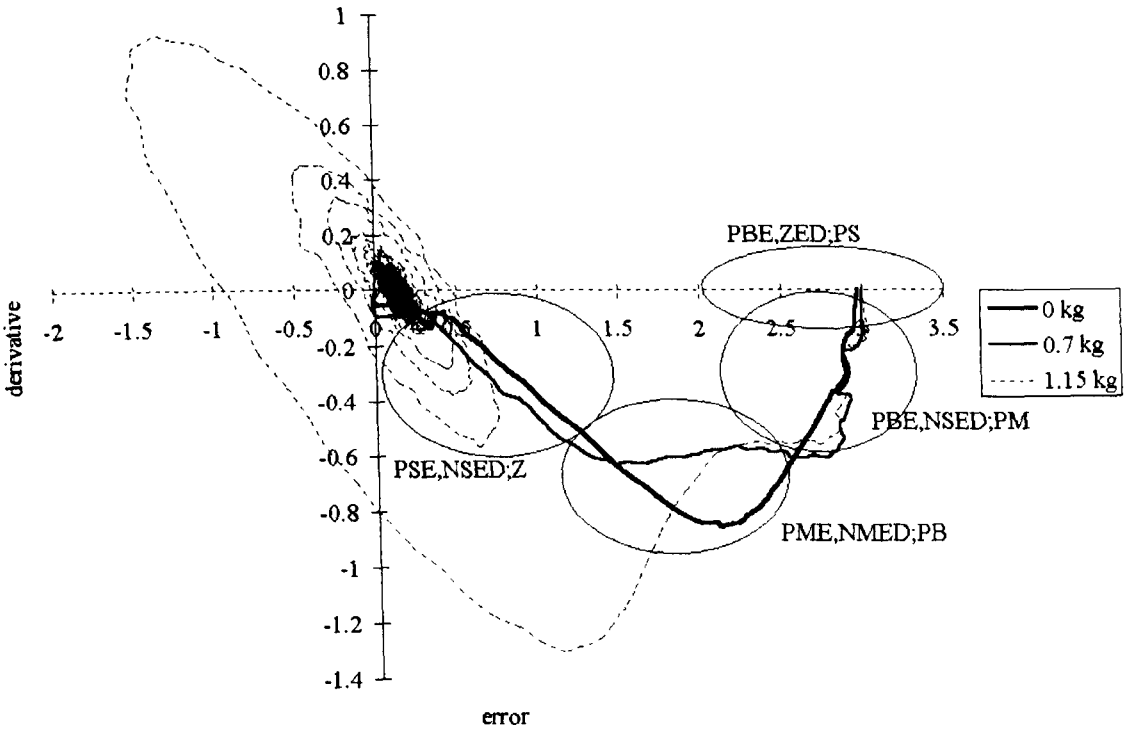

Figure 6.10: Measured linguistic trajectory for the fuzzy PD controller (different payloads)

The resulting linguistic trajectory for the 0 and 0.7 kg cases is:-

PS (4) → PM (6) → PB(8) → *no rule for PSE,NSED i.e.,* $K_d = 0$.

It can be seen for the 0 and 0.7 kg payload cases that the rules designed are consistent with the design and that there are no conflicting rules. The rulebase is not complete as there are no rules defined for the PS,NS part of the trajectory. This is not undesirable as at this stage the trajectory has already been influenced by the earlier rules and very little damping is required. This rule behaviour is still in line with the original design philosophy of having high initial damping and then turning it off when the trajectory is near the set point. It can also be observed that the rules in the top left, top right and bottom left quadrants have not been fired.

For the 1.15 kg case, the linguistic trajectory is initially similar to the 0 and 0.7 kg cases i.e., PS (4) → PM (6) → PB(8). But after this the trajectory follows a path where there are undefined rules i.e., there are no rules defined for PME,NBED and PSE,NBED and ZE,NMED. After this undefined stage the trajectory once again begins to follow a defined path of PM (6) → PS (4) . Also, there is no rule defined for NSE,PMED. It is quite likely, with re-design of the controller to take in account the more critical rules in the bottom right quadrant, that this rule and others in the bottom left and top right and left quadrants will not be fired as seen in the 0 and 0.7 kg cases. Taking this into observation into account and inclusion of the critical rules for the 1.15 kg case will result in a rulebase with a total of 7 rules in contrast to the 12 rules initially defined.

Table 6.5 summarises the performance of the variable payload fuzzy PD controller in response to a step input. Results for the PD controller are included for comparison.

Controller	K_p	K_d	$t_r(s)$	$t_s(s)$	$e_{ss}(mm)$	$M_{pt}(\%)$	Payload/(Kg)
fuzzy PD	0.6	variable	0.033	0.084	0.12	0	0
fuzzy PD	0.6	variable	0.033	0.064	0.15	0	0.7
fuzzy PD	0.6	variable	0.033	0.427	-	50	1.15
PD	0.6	4	0.018	0.09	0.31	17.3	0
PD	0.6	4	0.018	0.242	0.33	61.3	0.7

Table 6.5: Performance of the variable payload fuzzy PD controller

From table 6.5 it can be seen that the fuzzy PD controller gives a better performance in terms of overshoot and settling times for the 0 kg and 0.7 kg payload cases in comparison to PD control. The rise-time of the fuzzy controller however, is almost 50 % slower (0.033 s vs. 0.018 s). This observation is in agreement with the design philosophy applied to the practical fuzzy system since higher damping (via K_d) is applied at an earlier stage than the simulated fuzzy controller. However, simulations show that the rise-time could also be made superior to the fuzzy controller.

The steady state performance, e_{ss}, of the fuzzy PD controller has also improved over its classical counterpart by a factor of 2.58 (0.31 vs 0.12) for the no-load case and 2.2 (0.33 vs 0.15) for the 0.7 kg case. These steady state errors obtained are better than the ones achieved by Lewis [10] (0.45 mm - for unloaded arm).

Results are also shown for the 1.15 kg case under fuzzy control although the response is stable, the settling time has increased considerably from under 0.1 s to 0.43 s. Examination of the linguistic trajectory for this payload case reveals that the rules being applied are inappropriate as there are undefined rules (which result in no damping output) along the linguistic trajectory. It is likely that re-design of the rulebase bearing these anomalies in mind will yield a better performance for the 1.15 kg case. Re-design of the rulebase to include the undefined rules for the 0 and 0.7 kg and 1.15 kg case and pruning the un-fired rules could potentially reduce the number of initially designed rules from 12 to 7.

There was no direct comparison to PD control for the 1.15 kg case as the controller became unstable under classical control. This shows that, although there were some undefined rules in the rulebase, the influence of the fuzzy supervisor is beneficial to controller performance. All higher payload cases under either PD or fuzzy PD control also became unstable. The reason for the instability in the controller is a combination of higher payload and relatively high derivatives being applied to noisy measurement signals (the relatively high derivative amplifies the effects of noise in the control loop).

The practical fuzzy PD controller became unstable for the 1.6 kg case. This instability is not predicted by the simulation of the fuzzy PD controller with different payloads in section 5.6.3 where the 1.6 kg case is controlled with virtually no overshoot. The reason for this discrepancy is due to a couple of factors:-

- Inaccuracy of the model as identified in chapter 3.
- The simulation signal measurements are not as noisy as the practical system (caused by the use of noisy PWM amplifiers). Higher derivative terms required to damp to the higher payload make the control system even more sensitive to noise.

6.6.2 Robustness and stability

The fuzzy rules are designed in such a way as to maintain the derivative value low during operation and only inject higher values when more damping is required. Using this design philosophy it has been possible to design a controller that is capable of higher payload that remains stable while tracking and has an improved transient performance over classical PD control. Maintaining a low derivative in the controller serves to increase the stability margin of the controller hence increases the robustness of the system. The following sections briefly investigate the robustness of the fuzzy controller in the context of ill defined rules and external disturbance rejection.

Rule sabotage

Failure of all the fuzzy rules in the controller, simulated by deleting all the rules, reduces the controller to the simple classical PD controller. A more interesting case arises when 'rogue' or ill defined rules are designed. The matrix in Fig. 6.11 below gives an example of a rulebase where the rules designed along the linguistic trajectory have been altered to give a much higher damping output in the initial stages of the linguistic trajectory. Rules beginning with PBE and PME are most likely to be encountered in the initial stages of the linguistic trajectory hence it is these rules that have been changed to have higher outputs. Rules have been changed as follows:-

1. PBE,ZED;PS → PBE,ZED;PM
2. PBE,NSED;PM → PBE,NSED;PBB
3. PME,NMED;PB → PME,NMED;PBB

ė \ e	NS	Z	PS	PM	PB
PS	PS_4	PS_4	PS_4		
Z	PS_4	Z_0			PM_6
NS	PM_6	PS_4			PBB_{12}
NM	PB_8		PM_6	PBB_{12}	
NB					

Figure 6.11: Fuzzy rule base with sabotaged rules (many payload design)

Fig. 6.12 below depicts the resulting the transient performance of many-payload design where the fuzzy rules have been sabotaged.

It can be seen that there a definite degradation in performance from the response shown in Fig. 6.8. The effect of the very high initial derivative is clear. For the lower payload cases it can be seen that once the higher derivative is injected, (at 0.13s) the response appears to undershoot before settling down at $\approx$ 0.3s. The long period for undershoot reflects the duration of the application

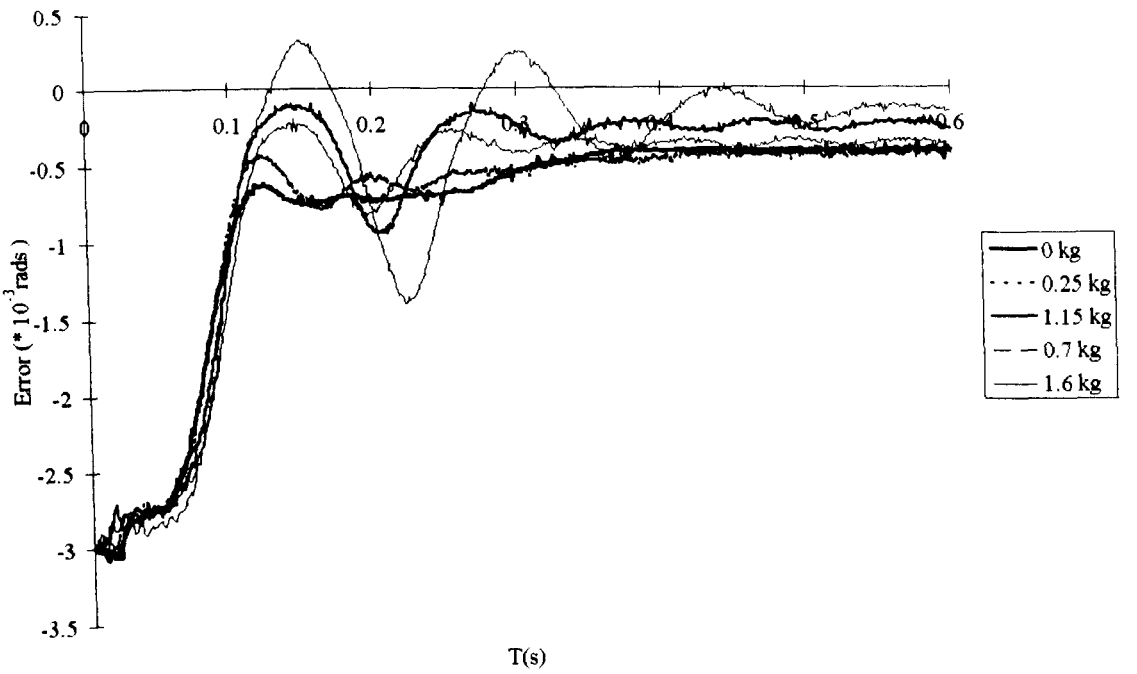

Figure 6.12: Transient response with sabotaged fuzzy rules (different payloads)

of the high derivative gain. Larger steady state errors ($\approx$ 0.17 mm) are also seen for the lower payload cases this is because the system was not given a chance to reach the desired set-point as a result of application of the very high damping. Also, with this sudden injection of high damping, there is a tendency for the response to become more oscillatory as the payload increases. With maximum payload, an overshoot of 10% is observed, however the undershoot is far greater ($\approx$ 50%), indicating that the rules are over compensating for overshoot.

Although the controller remains stable whilst tracking that there is greater ripple in the step response giving slower settling times. The no-payload case distinctly shows where the derivative has been highest along the linguistic trajectory. This is a very important observation, as it can be seen clearly how the transient response has been 'tailored' by adjusting the derivative gain. The observations in this section also highlight the fact that the rules for the controller need to be carefully designed.

Disturbance rejection

For disturbance rejection tests the arm was set at 0° elevation with a 4.2 kg payload attached. The fuzzy rulebase used in this case was similar to that for motion in the horizontal plane. The test was simply to cut the string holding the mass and to observe the closed loop response of the system to this massive disturbance. Figure 6.13 shows disturbance response under fuzzy PD control.

Although the fuzzy rule base has not been optimised for motion in the vertical plane it can be seen that the system remains stable under fuzzy PD control. Tip vibrations are damped more quickly than the uncontrolled version. The ratio of tip mass to the unloaded arm in this case:-

$$r = \frac{max\ payload}{mass\ of\ arm\ +\ mass\ of\ tip\ sensor} = \frac{4.2\ kg}{0.885\ kg\ +\ 0.15\ kg} = 4.042 \qquad (6.2)$$

If the mass of the sensor were considered to be part of the payload, then $r = 4.93$, i.e., almost 5 times the mass of the arm. The arm is brought to rest in 1.2 s under fuzzy control. This is 20 % quicker than the uncontrolled response of 1.6 s.

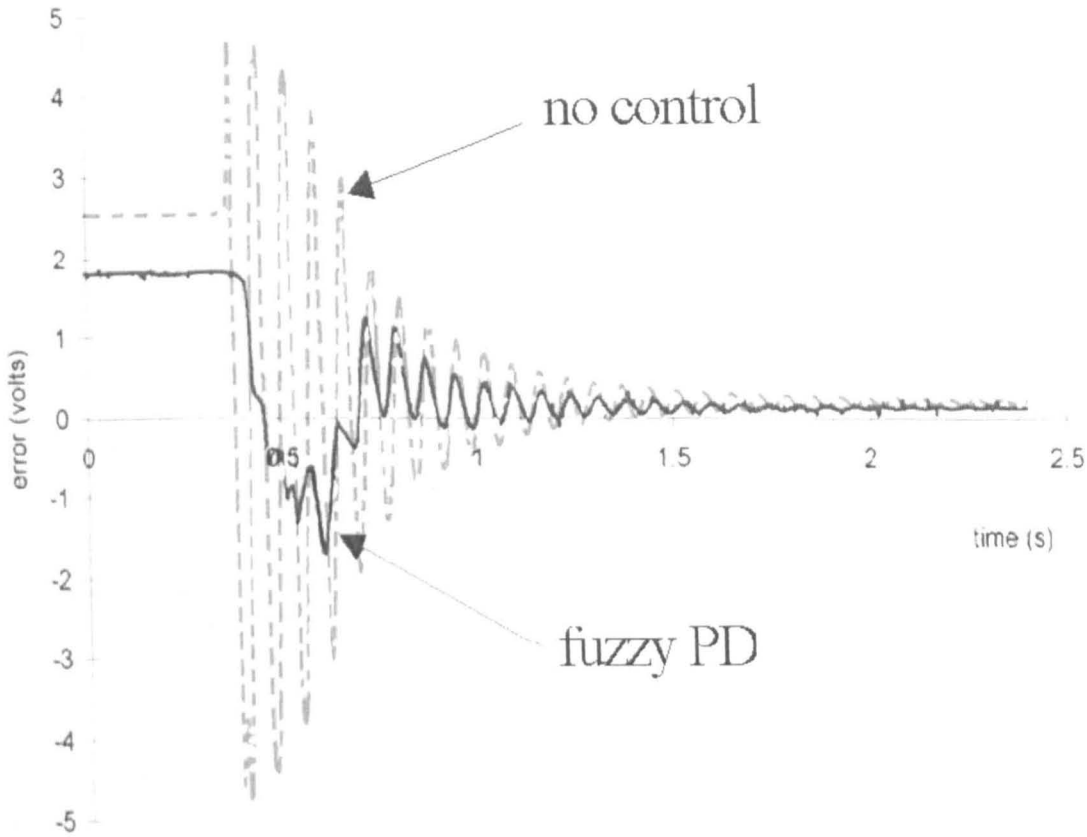

Figure 6.13: Fuzzy PD controller response to a massive disturbance.

6.7 Performance analysis - Processing

The previous section evaluated the performance of the fuzzy PD controller using the standard transient response criteria. This section evaluates the performance of the controller implementation in terms of processing. The controller performance is analysed with reference to real-time implementation on the transputers. The main criteria for performance evaluation are speedup, i.e., the ratio of increase in processing speed between the sequential and parallel implementation, and memory usage.

6.7.1 Speedup

The fuzzy PD controller was implemented on different hardware during development in a bid to achieve faster control sampling rates. Table 6.6 gives the processing times for the different implementations. Early implementations of the fuzzy controller utilised output consequent sets as a opposed to singletons. The processing times for different configurations of the consequents, i.e., implemented as output sets with $N_q = 50$ and 5 output singletons were also compared. The architectures for the T800 and T805 implementations are given in section 6.3.2.

Platform	Consequent	No. of rules	No. of axes	Control sample rate
PC	fuzzy set	9	1	10 ms
Transputer (T800)	fuzzy set	9	1	2 ms
Transputer (T805)	fuzzy set	14	2	1.2 ms
Transputer (T805)	5 singletons (semi-parallel)	14	2	0 .27 ms

Table 6.6: Achieved processing times for the different hardware configurations

It is clear from the observations above that the output singleton implementation gives the fastest controller sampling rate (a speed-up of 4 times over a similar implementation but with output fuzzy sets). A rough overall speed-up can be calculated for the fastest transputer implementation

and the slowest PC implementation (this calculation includes the AD and DAC cards - PC30B (@ 30 kHz) and University of Bangor custom made (@ $\approx$ 40 kHz)). Assuming that control of a second axis would double the processing time for the PC implementation:-

$$Speedup = \frac{PC\ processing\ time\ (9\ rules\ on\ each\ axis)}{T805\ processing\ time\ (14\ rules\ on\ each\ axis)} = 74 \tag{6.3}$$

6.7.2 Memory

The implementation of the fuzzy input lookup tables for a single set takes 50 x 7 = 350 bits (can be implemented using an input resolution of 6 bits and an internal resolution of 7 bits) as opposed to Surmann and Ungering's smallest memory implementation that took 255 x 8 = 2040 bits for an 8 bit input resolution and an internal resolution of 8 bits. This represents a saving in memory of 78% per fuzzy set. With antecedant arrays of 40 elements, the saving would increase to 84%.

6.8 Summary

The fuzzy controller has been implemented in a transputer environment. By placing the different controller functions i.e., inference, defuzzification and output processing on to separate concurrently running tasks and by use of output singletons, it has been possible to speed up the execution of the fuzzy controller. Sampling rates of up to 4220 kHz for simulataneous dual axis control were obtained through the transputer based approach. A rough speedup of 74 was obtained between a sequential dual axis PC implementation and the dual axis parallel transputer implementation. Semi-parallel processing of the inference stage using transputers was shown to yield a speed up of 1.92 over sequential processing.

The fuzzy PD algorithm has proved to be suitable for real-time control in an embedded environment. This was made possible by considering the design of the controller from a real-time implementation perspective. Fast processing was achieved configuring the fuzzy controller in the following manner:-

- Use of look-up table antecedant sets and encoding the membership functions as elements of 8 bit integer arrays for quick de-referencing.
- Use of integer based computation in the processing of the fuzzy rules.
- Use of *individual rule based* firing with *min* for conjunction of the input fuzzy sets and product for rule firing *scaled implication.*
- Use of weighted average defuzzification and output singleton sets.
- Implementation of the inherently parallel fuzzy algorithm and control for each separate axis on parallel processors. Transputers have been used to process the different stages of the algorithm i.e., inference, defuzzification and output processing. Of these processes, inference was found to have the highest processor overhead. Semi-parallel processing of groups of rules yielded a speed-up in the inference stage of 1.92 times. Further speed-up could potentially be obtained if the individual rules were processed completely in parallel. The 0.27 ms processing time can be further reduced to under 21 μs if the bottleneck due to I/O processing is addressed whilst retaining the same data acquisition cards. The optimised processing speed compares well with some highly optimised implementations that have been programmed in assembler.

The input resolution of the membership functions was found to greatly influence memory requirements of the fuzzy controller. Investigations were made to determine the most efficient resolution for the antecedant sets without compromising control performance. It was found that by encoding the antecedants in byte arrays of 40 or 50 elements each as opposed to array lengths of upto 256 elements (8 bit external resolution), reported in the literature, massive reductions in the storage requirement per fuzzy set could be made. This finding is especially significant when considering cheaper embedded μprocessor implementations that do not have large amounts of ROM available.

As a result of the efficient implementation of the fuzzy controller large amounts of bandwidth in terms of memory and processing speed are available on the transputers for further experiments with more complicated fuzzy based controllers with more rules based upon the same architecture.

The total number of rules required for the controller was 28 (14 for each axis), this number of rules is fewer (21 rules less than smallest implementation reported by Wu [12]) than the numbers reported by workers developing controllers for similar applications. Further reduction in the number of rules (7 per axis) can be achieved through pruning the rules that do not lie on the modified phase-plane linguistic trajectory.

The fuzzy PD controller outperforms classical control and meets with the performance specifications set in section 1.5.2. Fuzzy PD control has brought a marked improvement to the settling time and overshoot performance. The settling time has improved by upto 73% under payload and overshoot has been reduced from over 60 % to none. These represent significant improvement over the classical controller. The performance was degraded in terms of rise time by almost 84%. This degradation is in line with the design philosophy of 'slowing' the system down in the initial stages of the transient response.

The fuzzy PD controller has also improved on the classical controllers payload capacity. The transient response with a 1.15 kg tip mass is seen to be stable under fuzzy control but not so under classical control. This represents an increased payload capacity of over 64%.

The fuzzy PD controller was observed to be robust to sudden and massive external disturbances. The controller remained stable when a 4.2 kg (just under 500% of the link mass) payload was suddenly released by the arm. Vibrations were damped more quickly than the natural response of the system.

Sabotage of the fuzzy rules has an impact on the controller performance. A rule sabotage experiment where the damping terms in the early part of the linguistic trajectory were increased by over 50% showed degraded performance in rise time, settling time and steady state error. The controller remained stable for all payloads including 1.6 kg. This is an implied increase in payload capacity of 128% over PD control. However, significant undershoots (50%) were experienced for the higher payload cases of 1.15 and 1.6 kg.

References

[1] Surdhar J. S., White A. S, Stoker M., Gill R., and Korhonen J. A transputer implementation of a fuzzy PD controller applied to a flexible link manipulator. In *Proc. DARS'95 - Workshop on Human Oriented Design of Advanced Robotic Systems*, Vienna, Austria, 1995. ISBN 08 04 26042.

[2] Kosko B. *Neural Networks and Fuzzy Systems.* Prentice Hall, UK, 1992.

[3] Graham I. and King T. *The transputer handbook.* Prentice Hall, UK, 1990.

[4] Brookes G. and Stewart A. *Introduction to occam2 on the transputer.* Macmillian Education Ltd., 1989.

[5] Inmos. *The Transputer Databook.* Inmos, Bristol, U. K., second edition, 1989.

[6] Surmann H. and Ungering A. P. Fuzzy rule based systems on general purpose processors. *IEEE Micro*, pages 40–48, 1995.

[7] Costa A., De Gloria A., Farabischi P., Pagni A., and Rizzotto G. Hardware solutions for fuzzy control. *Proceedings of the IEEE*, 83(3):422–434, 1995.

[8] Boverie S., Demaya B., and Titli A. Fuzzy logic control compared with other control approaches. In *Proceeedings of the* 30^{th} *conference on decision and control*, pages 1212–1216, Brighton, England, 1991.

[9] Voit G. Fuzzy Logic in C. *Dr. Dobbs Journal*, 197:40–49, 1993.

[10] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[11] Rabiner L. R. and Gold B. *Theory and Application of Digital Signal Processing.* Prentice Hall, 1975.

[12] Wu J. and Tsuei Y. G. Comparison of fuzzy logic and self-tuning adaptive control of single-link flexible arm. *Mechatronics*, 3(4):451–464, 1993.

Chapter 7

Conclusions

This research has shown that fuzzy based control has been successfully applied to the control of a flexible manipulator with a round cross section. This is a significant contribution since the majority of previous research into direct end-point control has concentrated on the used of slender 'theoretical' structures that are relatively easier to model. This is also the first implementation of fuzzy logic to control a long reach robot in the TFS based configuration.

Other achievements of this research are:-

Development of a novel modified phase plane method The modified phase plane method has been successfully used to develop complete, consistent and non-interactive rules for a supervisor based fuzzy controller. The method is part of the design methodology investigated to develop a fuzzy controller that is suitable for application in a real time and safety critical environment.

Development of an efficient fuzzy algorithm A highly efficient fuzzy algorithm, suitable for real-time embedded control has been devised. The design of the fuzzy algorithm has focused on a real-time implementation from the outset. This approach provided justification for all the parameters of the fuzzy controller including the membership functions, inference and defuzzification methods and operators used in the algorithm. The algorithm is efficient both in terms of processing overhead, number of fuzzy rules and memory requirements for the antecedant sets.

Controller sampling times of 0.27 ms have been achieved for a dual axis controller. This processing speed is 74 times faster than the preliminary proof of concept sequential implementation. This sample rate is 55 times better than the 15 ms sample rate reported by Moudgal *et al.* [1] for a fuzzy control of a flexible link manipulator.

The pruned rulebase results in a total of 14 rules for dual axis control (i.e., 7 rules per axis). This number of rules is far fewer than then number of rules reported by other workers for a similar application (Wu and Tsuei [2] reported 49 rules for a single axis controller).

By using lower input resolutions and encoding the antecedant sets in byte arrays of 40 elements, savings of 84% per fuzzy set have been made over Surmann and Ungerings [3] most efficient implementation. These savings are especially significant when implementation of a fuzzy controller on a memory constrained device such as an embedded processor is considered.

Enhancement of the performance envelope of a classical PD controller Fuzzy PD control has been successfully applied to the optically guided long reach TFS based robot. By designing a fuzzy supervisor about a lower level classical PD controller it has been possible to enhance the performance envelope of classical PD control. These performance improvements are summarised below:-

- Payload adaptive non-linear control. By injection of high derivative gains during the early stages of step-like inputs and maintaining low derivative terms during steady state, it has been possible to kill off the high overshoots due to higher payload and maintain stable operation of the manipulator during tracking.

Stable transient control and tracking was achieved for a maximum of 1.6 kg, i.e., a payload to link mass ratio of $r = 1.8$. This ratio represents an improvement in payload capacity of 128% over classical PD control.

The ratio, r, also compares favourably with the reported results of Rovner and Cannon [4] ($r = 1.5$), Wu *et. Al.* [2] ($r = 2$) and Takahashi and Yamada [5] ($r = 2.5$). TFS based fuzzy control differs fundamentally from these cited results in that the additional effects of link torsion and gravity are also prevalent. Also, the order of the payloads used in the reported experiments were almost a factor of ten lighter than those used on the TFS based robot (maximum 250 g vs. maximum 1.6 kg).

- Better transient performance over classical PD control. The improvements in transient performance are summarised in the table below. P_m represents the performance measure and a '$\surd$' (tick) in the *Note* column denotes a positive improvement as opposed to an 'X' which denotes a degradation in performance.

Payload/kg	P_m	PD	Fuzzy	Improvement (%)	Note
0	$t_s(s)$	0.09	0.084	6.6	$\surd$
0	$t_r(s)$	0.018	0.033	83.3	X
0	$e_{ss}(mm)$	0.31	0.12	61.2	$\surd$
0	$M_{pt}(\%)$	17.3	0	100	$\surd$
0.7	$t_s(s)$	0.242	0.064	73.5	$\surd$
0.7	$t_r(s)$	0.018	0.033	83.3	X
0.7	$e_{ss}(mm)$	0.33	0.15	54.5	$\surd$
0.7	$M_{pt}(\%)$	61.3	0	100	$\surd$
1.15	$t_s(s)$	-	0.427	-	$\surd$
1.15	$t_r(s)$	-	0.033	-	$\surd$
1.15	$e_{ss}(mm)$	-	-	-	-
1.15	$M_{pt}(\%)$	-	50	-	$\surd$

Table 7.1: Summary of the performance of the fuzzy PD controller.

Table 7.1 shows that fuzzy PD control has improved on the performance envelope of the classical PD controller in terms of settling time (t_s), steady state error (e_{ss}) and peak overshoot (M_{pt}). These improvements have come at the expense of having a slower rise time (t_r) which is in line with the design philosophy for the variable payload controller.

- Greater robustness. The design philosophy of the fuzzy controller is to maintain the derivative terms of a classical PD controller at a low level during steady state and during tracking. Higher derivative terms are only injected when large overshoots need to be damped. This philosophy is in tune with enhancing the robustness of the controller since the controller is susceptible to instability due differentiation of noise from the PWM amplifiers used.

  The supervisor based topology makes the fuzzy PD controller robust to failure of all the fuzzy rules. In this case the controller reduces to the classical PD controller. Rule sabotage along the linguistic trajectory of fuzzy controller results in degraded performance in the controller. Sabotage of rules on the linguistic trajectory have the greatest impact on performance, resulting in high undershoots and large steady state errors. However, the controller is never seen to go unstable.

  The fuzzy PD controller is robust to sudden massive external disturbances and is able to damp vibrations due to a sudden step in payload (from 475% of the arm mass to 0%) more quickly (by 20%) than the natural response of the flexible manipulator arm.

References

[1] Moudgal V. G., Kwong W. A., Passino K. A., and Yurkovich S. Fuzzy learning control for a flexible link robot. *IEEE Transactions on Fuzzy Systems*, 3(2):199–210, 1995.

[2] Wu J. and Tsuei Y. G. Comparison of fuzzy logic and self-tuning adaptive control of single-link flexible arm. *Mechatronics*, 3(4):451–464, 1993.

[3] Surmann H. and Ungering A. P. Fuzzy rule based systems on general purpose processors. *IEEE Micro*, pages 40–48, 1995.

[4] Rovner D. M. and Cannon R. H. Jr. Experiments toward on-line identification and control of a very flexible one- link manipulator. *The International Journal of Robotics Research*, 6(3):3–19, 1987.

[5] Takahashi K. and Yamada I. Neural network based learning control of flexible mechanism with application to single-link flexible arm. *Transactions of the ASME*, 116:792–795, 1994.

Chapter 8

Recommendations for further work

The recommendations for further work can be separated into two categories, i.e., those involving improvements in the prototype robot design and structure and those related to the modelling and control of the robot.

8.1 Improvements to the manipulator design

The prototype rig has been useful as a proof of concept of TFS based control. However, there are design anomalies that can be addressed in order to improve control performance.

8.1.1 Better robot structure

Chapter 3 described three different methods to measure the frequency response of the manipulator. It was found that much of the actual flexibility of the structure was due to the Harmonic Drive gears, the root joint bracket in the arm and the thrust bearing upon which the robot is supported facilitating motion in the horizontal plane. It was also noted that a possible source of torsion vibrations was the small diameter (8 mm) pivot dowel used for support of the root joint bracket on the opposite side of the vertical axis actuator.

As a result of the above studies the following recommendation are put forward:-

- Replacement of the simple thrust bearing with a more stable bearing.
- Use of a single process milled structure for the link as opposed to being bolted to the root joint bracket.
- Use of a support shaft with a larger cross sectional area in order to increase its load bearing capacity.
- Stiffening of the actuator supporting frames.

8.1.2 Extension to multiple links

Lewis carried out studies into the extension of the TFS based control configuration to multiple axes. Further details can be found in [1] and [2]. These ideas need to be implemented practically.

8.2 Modelling and controller improvements

The author briefly investigated adaptive fuzzy control and neural network control. It is possible that further improvements in performance can be made using such methods. These areas need to be investigated more thoroughly.

8.2.1 Improvements to current model

For model based control the ACSL model requires the following improvements:-

- Inclusion of gravity terms and the effects of coulomb friction.
- More accurate modelling of the flexure caused due to the combination of harmonic drive gearing, toothed gearing and manipulator root joint bracket.
- Inclusion of backlash in the gearing.
- Inclusion of more sensors on the system for better identification of the plant. However, this later point detracts from the advantages of TFS based control which only requires a single sensor/detector arrangement.

8.2.2 Derivation of a AI based model

A model based on ANN's could be investigated with a view to simulating the TFS based robot. This method of modelling could be a viable alternative to a description of the system using dynamic equations. It is possible that this model will be more accurate as the model could be based on training data taken from measurements on the real robot. There will be little concern about the size of the model (memory requirement) or time it takes to train the ANN since the model will only be used offline for evaluation of the fuzzy control rules.

8.2.3 Fuzzy control

Further investigations can be carried out in the following areas:-

- Adaptive fuzzy control. A method of making the fuzzy PD controller adaptive via the Least Mean Squared (LMS) algorithm has been briefly investigated. The initial results obtained did not show a difference in performance between the non-adaptive and adaptive versions. It was suspected that the adaptation parameters for the learning algorithm may not have been chosen correctly. Further investigations need to be carried out to determine the correct learning coefficient for the LMS algorithm investigated. Also other methods of learning may yield better results. These need to be investigated bearing in mind a real-time application.

  Other parameters of the fuzzy controller can be adjusted to improve control performance and if adapted online could provide a means of making the fuzzy algorithm more adaptive or 'self-learning'. The control experiments in this thesis only considered adjusting the rule parameters and input resolution in order to make the fuzzy algorithm adaptive and to give optimised performance. Adjustment of the membership parameters such as set shape and degree of overlap were not investigated. It is possible that improvements in controller performance can be made if the configuration of these parameters is optimised, for instance increasing to overlap between sets has been shown to reduce steady state error in the response of a flexible manipulator [3]. The trade-off in performance between the choice of an optimal overlap ratio, $T_x = 1$ as defined by Bolinger, and other ratios of overlap will also have to be investigated.

  Rudas and Kaynak [4] studied the effect of fuzzy operators on the quality of control. They proposed new generalised intersection and union operators which could be constructed from the conventional *max* and *min* T-norm operations. They found that the performance of the fuzzy controller could be improved using the modified operations for a class of plants such as first and second order non-linear plants. The operators used in the fuzzy controller of this thesis were chosen with processing speed in mind (*min* represented by a product and *max* represented by a sum). It would be interesting to examine the difference in performance in the controllers and to evaluate it with respect to trade-off in processing speed due to the greater number of computations required to implement the modified operators.

- Other fuzzy paradigms. Palm [5] and Yi and Chung [6] investigated methods of achieving robust fuzzy control through appropriate rule design. Rules for their controllers were designed according to upper-diagonal-negative lower-diagonal-positive (UNLP) pattern. This would mean that all the output controller values in the upper diagonal of the rulebase in Fig. 6.7 would have to be negative and increasing proportionally with the size of positive error. The lower diagonal rules would have to be symmetric and opposite i.e., positive outputs from the controller increasing as the error gets more negative. Design of fuzzy rules in this way yields a controller that is an extension of variable structure control with a boundary layer. Such fuzzy controllers are especially desirable for the control of non-linear dynamic systems with uncertainty, especially in DF control. The SF controller in this thesis can easily be adapted to this type of control by changing the rulebase to yield the desired outputs. Further experiments need to be carried out to examine the performance of UNLP based rule design on the TFS based long reach manipulator.

  It has been mentioned that further robustness can be incorporated into the controller via design of exception handling rules in the fuzzy controller. These rules can be designed in such a way as to force the controller to be stable. One such way to do this would be to populate the current un-defined cells in the fuzzy controller matrix with rules that follow the UNLP criterion. In this way the performance of the controller is not compromised with regard to transient performance and an additional element of robustness is added to the control system.

8.2.4 Neural network control

The author briefly investigated a neural network approach to the control of the TFS based arm [7]. A multi-layer perceptron (MLP) neural network was shown to be less noisy and provide better control, in terms of overshoot and steady state error, when tested with a 500 g payload in comparison to a fuzzy PD controller (designed for no payload case) and Radial Basis Function (RBF) neural network. Although the topology of the network was optimised (five processing elements in a single hidden layer) in terms of the number of processing elements by using cascade correlation [8], the training set was not universal for all configurations of the manipulator (only data from a RHS step was used in training). This latter issue and a safe architecture for implementation needs devised before the neural network approach is applied to the TFS based robot in practice.

8.2.5 Optimisation of processing

Although the control sample rates achieved using the semi-parallel transputer based approach are more than adequate for the control of the application described in this thesis, further speed in processing can be achieved using the methods suggested in section 6.3.2. The extra bandwidth can be used to investigate other control algorithms that have a high processing overhead such as optimal control using some form of fuzzy or neural network payload estimation scheme for identification.

The controller implementations described in this thesis had a relatively small number of fuzzy rules (28 rules for dual axis control). In these implementations all the rules are evaluated in order to compute an inference. In the case of a large number of rules this would be inefficient in terms of processing speed. In order to reduce this overhead it would be necessary to adopt schemes such as Surmanns [9] and Costa [10] to evaluate only a fixed number of rules for and arbitrarily large rulebase. Alternatively, based on the adaptive fuzzy implementation described in this thesis, it would be possible to prune rules which fall below a certain pre-defined activation threshold or confidence (defined by the strength of the adapted weight attached to each rule). This dynamic approach may be more efficient than evaluating a fixed number of rules.

8.2.6 Improvement of the control environment

The PWM amplifiers used for driving the actuators are not suitable for controllers with high value derivative terms. Amplifiers which are less noisy should be obtained.

References

[1] Lewis J. The design, construction and testing of a laser guided robot arm, MSc dissertation, Middlesex University, London, England. 1991.

[2] Lewis J. *A Steady State Tip Control Strategy for Long Reach Robots.* PhD thesis, Middlesex University, London, England, 1996.

[3] Lee J., Vukovich G., and Sasiadek J.Z. Fuzzy control of a flexible link manipulator. In *Proceedings American Control Conference, Baltimore, Maryland*, pages 568–574, 1994.

[4] Rudas I. and Kaynak M. O. Entropy based operations on fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 6(1):33–40, 1998.

[5] Palm R. Sliding mode fuzzy control. In *Proceedings of the IEEE* 1^{st} *International Conference on Fuzzy systems*, pages 519–526, 1992.

[6] Yi S. Y. and Chung M. J. Robustness of fuzzy logic control for an uncertain dynamic system. *IEEE Transactions on Fuzzy Systems*, 6(2):216–225, 1998.

[7] Surdhar J. S., White A. S., and Gill R. Neural network control applied to a flexible link manipulator in a tip feedback sensor based (TFS)based configuration. In *Proc. of the* 12^{th} *International Conference on CAD/CAM Robotics and Factories of the Future*, London, UK, 1996.

[8] Fahlman S. E. and Lebierre C. The cascade-correlation learning architecture. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, pages 183,304. Morgan Kaufmann, San Mateo, CA., 1990.

[9] Surmann H. and Ungering A. P. Fuzzy rule based systems on general purpose processors. *IEEE Micro*, pages 40–48, 1995.

[10] Costa A., De Gloria A., Farabischi P., Pagni A., and Rizzotto G. Hardware solutions for fuzzy control. *Proceedings of the IEEE*, 83(3):422–434, 1995.

Appendix A

ACSL main file - flexible manipulator model (MODEL.CSL)

```
PROGRAM ROBOT ARM SIMULATION

! all the parameters are correct as specified by the data sheets
! we use a gear efficiency of 60%
! 6/12/96
! calculation of sets using sets.cpp


!macro definitions ----------------------------------

INCLUDE 'MACRO.IN'

!include sets ----------------------------------------

!INCLUDE 'E50.IN'

INCLUDE '50N.IN'
!INCLUDE '40N.IN'
!INCLUDE '30N.IN'
!INCLUDE '20N.IN'
!INCLUDE '10N.IN'


!initialisation of the program ----------------------
INITIAL

!new variables from fuzzy program
INTEGER I,J,P,N,AV,AVDTEST,FSW,CSW,ASIZE
REAL E,CT
ARRAY Y(7),V(7),M(7),D2Y(7),VIC(7),YIC(7),DY(7)
!set the number of stations to 17
!ARRAY Y(17),V(17),M(17),D2Y(17),VIC(17),YIC(17),DY(17)


!arm ------------------------------------------------

CONSTANT EI=8004,...
EIROOT=4001,...
L=1.18,...
N=7,...
MASS=0.1564,...
W=0.76

! changing the value of W has some effect on stability

!system----------------------------------------------
CONSTANT ANGVIC=0.0,...
THIC=0.0,...
TZ=0.0,...
VI=0.0,...
W1=0.1,...
T1=0.001,...
K=0.003,...
TSTP=1.5,...
K1=10.,...
```

```
P1=10

!motor ------------------------------------------------

CONSTANT KE=20.53,...
MU=0.00014,...
JM=0.00012,...
KT=0.26,...
R=3.4,...
NR=580,...
PCLIM=1.9,...
NCLIM=-1.9


!control ----------------------------------------------

CONSTANT KTIP=1,...
KD=4,...
KP=0.6,...
KI=0,...
CKD=4,...
FSW=1,...
CSW=0

!fuzzy initial weights---------------------------------

WGT1 = 1
WGT2 = 1
WGT3 = 1
WGT4 = 1
WGT5 = 1
WGT6 = 1
WGT7 = 1
WGT8 = 1
WGT9 = 1
WGT10 = 1
WGT11 = 1
WGT12 = 1
WGT13 = 1

!----------------system-----------------------------------------

CINTERVAL CINT=0.0015
MAXTERVAL MAXT=0.0001

NSTEPS NSTEP=1
DX=L/(N-2)
J=1

I1..CONTINUE
IF(J.GT.N)GO TO I2
VIC(J)=0.0
YIC(J)=0.0
Y(J)=0.0
M(J)=0.0
ANGVEL=0.0
ANGVD=0.0
J=J+1
GO TO I1

I2..CONTINUE
END

DYNAMIC
DERIVATIVE
       PROCEDURAL(D2Y,ANGVD=Y,M,ANGVEL,tipfb,THETA)

            THC=K*STEP(TZ)

            E=THC-tipfb

            DDE=DERIVT(0.0,E)
            IE=INTEG(E,0.0)

            INCLUDE 'thinfuz.in'
```

```
                              !control loop
                              VOLT=FSW*(KP*E+KD*DDE)+CSW*(KP*E+CKD*DDE)
                              CURRNT=((VOLT-(KE*ANGVEL))/R)

              !include motor current limits (rated)
              IF(CURRNT.LT.NCLIM) CURRNT=NCLIM
              IF(CURRNT.GT.PCLIM) CURRNT=PCLIM

          TORQUE=KT*CURRNT

          ANGVD=(TORQUE/(JM*NR))+(EI*M(2)/NR**2*JM)-(MU*ANGVEL/JM)


          D2Y(1)=(-M(2))/(W*DX**2)
          D2Y(N)=0.
          Y(1)=0.0
          Y(2)=0.0
          I=2

      L1..CONTINUE
           IF(I.EQ.2) THEN
             GO TO L11
           ELSE
             GO TO L111
           END IF

      L11..CONTINUE
      M(I)=EIROOT*(Y(I-1)-2*Y(I)+Y(I+1))/(DX**2)
           I=I+1
           GO TO L111

      L111..CONTINUE
           IF(I.GT.N-2)GO TO L2
           M(I)=EI*(Y(I-1)-2*Y(I)+Y(I+1))/(DX**2)
           I=I+1
           GO TO L1
      L2..P=2
       M(N)=-(M(N-2)*MASS)/(3.*MASS+(3.*W*DX))
           M(N-1)=-M(N)

      L3..CONTINUE
           IF(P.GT.N-1)GO TO L4
           X=(P-1.5)*DX
           D2Y(P)=((2*M(P)-M(P+1)-M(P-1))/(W*(DX**2)))-(X*ANGVD)
           P=P+1
           GO TO L3

      L4..CONTINUE

   END !PROCEDURAL

        DEFL=1.5*Y(N-1)-0.5*Y(N-2)
        V=INTVC(D2Y,VIC)
        CALL XFERBR(DY=V,N)
        Y=INTVC(DY,YIC)
        YDEG = ASIN(DEFL/L)

        !convert to radians
        DELTHETA = YDEG * 0.01745

        ANGVEL=INTEG(ANGVD,ANGVIC)

        !include tipfb parameter
        TIPFB=INTEG(ANGVEL,THIC)-(KTIP*DELTHETA)

END !DERIVATIVE

TERMT(T.GE.TSTP)

END !DYNAMIC

END !PROGRAM
```

A.1 ACSL utility macro (MACRO.IN)

```
!macro definitions ------------------------------------


!macro for initialisation of arrays to zero

MACRO MACRO INITIALISE(P)
MACRO RELABEL L330
MACRO REDEFINE Z
         PROCEDURAL(P(1))
            DO L330 Z=1,50
                P(1)(Z) = 0.0
         L330..CONTINUE
         END!PROCEDURAL
MACRO END


!macro for multiplication of array by scaler value
!NB the scalar multiplication variable is divided by 100
!so that we get a value between 0 and 100 as defiend in our sets
!otherwise we get some spurious behaviour!!

MACRO MACRO SMULT(P,Q)
MACRO RELABEL L330
MACRO REDEFINE H
         PROCEDURAL(P(1)=P(2),P(3))
            DO L330 H=1,50
                P(1)(H)=(P(2)/100)*P(3)(H)
         L330..CONTINUE
END !PROCEDURAL
MACRO END


!macro to calculate the minimum of the elements of two arrays

MACRO MACRO MINT(P,Q,R)
         PROCEDURAL(P(1)=P(2),P(3))
            IF(P(2).LT.P(3)) THEN
                P(1)=P(2)
            END IF
            IF(P(3).EQ.P(2)) THEN
                P(1)=P(3)
            END IF
          END!PROCEDURAL
MACRO END


!macro to compute the LMS

MACRO MACRO LMS(P,Q,R,S)
          PROCEDURAL(P(1)=P(2),P(3),P(4))
             IF((P(2)*P(2)).GT.(P(3)*P(3))) THEN
                 P(1)=P(1) + 2*0.3*P(2)*P(4)
             ELSE
                 P(1)=P(1)
             END IF
           END!PROCEDURAL
MACRO END


!example LMS call

!LMS(WGT = E, OLD_E, OLD_WGT)
```

A.2 ACSL fuzzy inference engine (THINFUZ.IN)

```
!scale input ranges and put in safety ranges ----------

          !max e = +/- 3*10e-3
```

```
          AV=(E*167*ASIZE)+(ASIZE/2)
             IF(AV.LT.1) AV=1
             IF(AV.GT.ASIZE) AV=ASIZE-1

          !max dde = +/- 6*10e-3
          AVDTEST=(DDE*8.3*ASIZE)+(ASIZE/2)
             IF(AVDTEST.LT.1) AVDTEST=1
             IF(AVDTEST.GT.ASIZE) AVDTEST=ASIZE-1


!output sets ----------------------------------------

PL  = 0
PS  = 4
PM  = 6
PB  = 8
PBB = 12


!fuzzy rules ----------------------------------------

          MINT(MIT1 = NSE(AV),PSD(AVDTEST))
          OMIT1 = MIT1 * PM * WGT1
          !LMS(WGT1 = E, ELAST, MIT1)


          MINT(MIT2 = NSE(AV),ZD(AVDTEST))
          OMIT2 = MIT2 * PL * WGT2
          !LMS(WGT2 = E, ELAST, MIT2)


          MINT(MIT3=NSE(AV),NSD(AVDTEST))
          OMIT3 = MIT3 * PS * WGT3
          !LMS(WGT3 = E, ELAST, MIT3)


          MINT(MIT4 = ZE(AV),ZD(AVDTEST))
          OMIT4 = MIT4 * PL * WGT4
          !LMS(WGT4 = E, ELAST, MIT4)


          MINT(MIT5 = ZE(AV),NMD(AVDTEST))
          OMIT5 = MIT5 * PB * WGT5
          !LMS(WGT5 = E, ELAST, MIT5)


          MINT(MIT6 = ZE(AV),PSD(AVDTEST))
          OMIT6 = MIT6 * PL * WGT6
          !LMS(WGT6 = E, ELAST, MIT6)


          MINT(MIT7 = PSE(AV),NMD(AVDTEST))
          OMIT7 = MIT7 * PBB * WGT7
          !LMS(WGT7 = E, ELAST, MIT7)


          MINT(MIT8 = ZE(AV),NBD(AVDTEST))
          OMIT8 = MIT8 * PBB * WGT8
          !LMS(WGT8 = E, ELAST, MIT8)


          MINT(MIT9 = PME(AV),NMD(AVDTEST))
          OMIT9 = MIT9 * PL * WGT9
          !LMS(WGT9 = E, ELAST, MIT9)


          MINT(MIT10 = PME(AV),NBD(AVDTEST))
          OMIT10 = MIT10 * PL * WGT10
          !LMS(WGT10 = E, ELAST, MIT10)


          MINT(MIT11 = PBE(AV),ZD(AVDTEST))
          OMIT11 = MIT11 * PL * WGT11
          !LMS(WGT11 = E, ELAST, MIT11)
```

```
        MINT(MIT12 = PBE(AV),NSD(AVDTEST))
        OMIT12 = MIT12 * PL * WGT12
        !LMS(WGT12 = E, ELAST, MIT12)


        MINT(MIT13 = PBE(AV),NMD(AVDTEST))
        OMIT13 = MIT13 * PL * WGT13
        !LMS(WGT13 = E, ELAST, MIT13)


        MINT(MIT14 = PBE(AV),NBD(AVDTEST))
        OMIT14 = MIT14 * PL * WGT14
        !LMS(WGT14 = E, ELAST, MIT14)

!defuzzification ------------------------------------

        CT = OMIT1+OMIT2+OMIT3+OMIT4+OMIT5&
        &+OMIT6+OMIT7+OMIT8+OMIT9+OMIT10&
        &+OMIT11+OMIT12+OMIT13+OMIT14

        BKD = MIT1+MIT2+MIT3+MIT4+MIT5&
        &+MIT6+MIT7+MIT8+MIT9+MIT10&
        &+MIT11+MIT12+MIT13+MIT14

        !check for divide by zero error
        IF (CT .EQ. 0) THEN
           CT = 1
        ELSE
           CT = CT
        END IF


        IF (BKD .EQ. 0) THEN
           KD = 1
        ELSE
           KD =(CT)/(BKD)
        END IF
```

A.3 50N.IN

```
!dimension variables ---------------------------------

DIMENSION PBE(50),PME(50),PSE(50),ZE(50),NSE(50),NBD(50),NMD(50),...
NSD(50),ZD(50),PSD(50)

!error -----------------------------------------------


CONSTANT NSE = 100,100,100,100,100,100,100,100,100,100,...
100,100,100,100,100,100,100,100,100,100,...
100,99,74,49,24,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT ZE = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,25,50,75,99,74,49,24,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT PSE = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,25,50,75,100,...
100,100,100,99,79,59,39,19,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT PME = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,20,40,60,80,100,100,...
```

```
100,100,99,74,49,24,0,0,0,0

CONSTANT PBE = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,25,50,75,100,100,100,100

!derivative ----------------------------------------

CONSTANT NBD = 100,100,100,100,100,75,50,25,0,0,...
100,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT NMD = 0,0,0,0,0,24,49,74,99,100,...
100,100,100,79,59,39,19,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT NSD = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,20,40,60,80,100,100,100,...
100,99,74,49,24,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT ZD = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,25,50,75,99,74,49,24,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0

CONSTANT PSD = 0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,0,0,0,0,...
0,0,0,0,0,0,25,50,75,100,...
100,100,100,100,100,100,100,100,100,100,...
100,100,100,100,100,100,100,100,100,100


ASIZE = 50
```

Appendix B

C++ source file for computation of fuzzy sets (sets.cpp)

```
/*********************************************************
* program to calculate the fuzzy sets
* local defines changed according to type
* set ie error or derivative (dependent on their range)
*
* last revision 7/6/98
*********************************************************/

#include <iostream.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

/*********************************************************
* local defines
*********************************************************/


#define MULTIPLIER 167
int  array_size=10;  /* default value - program prompts for other values*/


/*********************************************************
* local variables
*********************************************************/

static int *index_array[40],*global_array;
double *input_set_array[40];


/*********************************************************
* function prototypes
*********************************************************/

/*compute the fuzzy sets*/
void Calc(int, int , int, int);

/*map the 4 valuse to array indexes*/
void error_range(int);

/*round off values between 0 and 1 (alternative to ceil or floor) */
int round_off(double f);


/*********************************************************
* Calc
*
*
*********************************************************/

void Calc(int a, int b, int c, int d)
{
   double slope_a = 0.0,
   slope_c        = 0.0,
   *x_array,
```

```
range           = 20,
incr            = 0.0,
dx              = 0.0;

int *y;

//dynamic allocation
x_array = (double*)calloc(array_size,sizeof(double));
y = (int *)calloc(array_size,sizeof(int));

     //range is from -ve to +ve
     incr=range/array_size;

     for(int i=0;i<array_size;i++)
     {
         x_array[i]=dx;
         dx += incr;
     }

 slope_a = 100/((b*range/array_size)-(a*range/array_size));
 slope_c = 100/((d*range/array_size)-(c*range/array_size));

     for(int p=0;p<array_size;p++)
     {
         if (x_array[p]<=(a*range/array_size))
         {

            y[p]=0;

         }

         else if (x_array[p]>=(a*range/array_size) &&
                 x_array[p]<=(b*range/array_size))
         {

            y[p]= (int)(slope_a * (x_array[p]-(a*range/array_size)));

         }

         else if (x_array[p]>=(b*range/array_size) &&
                  x_array[p]<=(c*range/array_size))
         {

            y[p]=100;

         }

         else if (x_array[p]>=(c*range/array_size) &&
                  x_array[p]<=(d*range/array_size))
         {

             y[p]= (int)(slope_c*((d*range/array_size)-x_array[p]));

         }

         else
           y[p]=0;
       }

       for(int v=0; v<array_size; v++)
       {
          global_array[v] = y[v];
       }

   //give back to heap
   free(y);
   free(x_array);

}

/*************************************************************
* round_off
*
*
*************************************************************/
```

```
int round_off(double f)
{
   int out_integer;
   double integer,fraction;

   fraction=(modf(f,&integer));

   if (fraction>=.5)
      out_integer=(int)ceil(f);
   if (fraction<.5)
      out_integer=(int)floor(f);

   return out_integer;
}

/*********************************************************
* error_range
*
*
*********************************************************/

void error_range(int error_loop)
{

   double ai,bi,ci,di;

   ai=*(input_set_array[error_loop])/1000;
   bi=*(input_set_array[error_loop]+1)/1000;
   ci=*(input_set_array[error_loop]+2)/1000;
   di=*(input_set_array[error_loop]+3)/1000;

   index_array[error_loop] = (int*) calloc(4,sizeof(int));

   *(index_array[error_loop])
   = round_off((double)((ai*MULTIPLIER*array_size)+ array_size/2));

   *(index_array[error_loop]+1)
   = round_off((double)((bi*MULTIPLIER*array_size)+ array_size/2));

   *(index_array[error_loop]+2)
   = round_off((double)((ci*MULTIPLIER*array_size)+ array_size/2));

   *(index_array[error_loop]+3)
   = round_off((double)((di*MULTIPLIER*array_size)+ array_size/2));

}

/*********************************************************
* main program
*
*
*********************************************************/

void main()
{
   FILE *fp;
   char f_name[40];
   int number_of_sets, *ptr_array[10];
   static int loop=0;
   int how_many_times=0;
   double left,left_middle,right_middle,right;


   printf("please input the number of data set samples\n");
   cin>>how_many_times;

   printf("\nPlease input the number of fuzzy sets for each sample:\n");
   cin>>number_of_sets;

   for(int count_of_sets=0; count_of_sets < number_of_sets; count_of_sets++)
   {
      printf("please input the array values\n");
      cin>>left >>left_middle >>right_middle >>right;

         input_set_array[count_of_sets] = (double*)calloc(4,sizeof(double));
```

```
        *(input_set_array[count_of_sets])   = left;
        *(input_set_array[count_of_sets]+1) = left_middle;
        *(input_set_array[count_of_sets]+2) = right_middle;
        *(input_set_array[count_of_sets]+3) = right;
  }


  for(int no_of_files=0; no_of_files < how_many_times; no_of_files++)
  {

        printf("\nplease input the array size\n");
        cin>>array_size;

        //allocate space for global array depending on array size
        global_array = (int *)calloc(array_size, sizeof(int));

        //loop for number of sets
        for (loop=0; loop < number_of_sets; loop++)
        {
           //now calculate the array indexes for each set in a given sample
           //loop must be declared static global
          error_range(loop);

            //use the indexes in the calculation of the sets
            Calc(*(index_array[loop]),
                    *(index_array[loop]+1),
                    *(index_array[loop]+2),
                    *(index_array[loop]+3));

           //free the array used to hold the indexes for next time in the loop
             free(index_array[loop]);

           //allocate space for an array pointed to by ptr_array[loop]
           ptr_array[loop] = (int *) calloc(array_size,sizeof(int));

           //fill the array with the computed set values
for(int k=0; k<array_size;k++)
{
               *(ptr_array[loop]+k) = global_array[k];
 }

         }

               printf("please input the filename\n");

              gets(f_name);

              fp = fopen(f_name, "w+");

               for (int q=0;q<array_size;q++)
               {
              //output to console
              printf("%d\t%d\t%d\t%d\t%d\n",\
                        *(ptr_array[0]+q),\
                        *(ptr_array[1]+q),\
                        *(ptr_array[2]+q),\
                        *(ptr_array[3]+q),\
                        *(ptr_array[4]+q));
                  //output to file
                  fprintf(fp, "%d\t%d\t%d\t%d\t%d\n",\
                                        *(ptr_array[0]+q),\
                                        *(ptr_array[1]+q),\
                                        *(ptr_array[2]+q),\
                                        *(ptr_array[3]+q),\
                                        *(ptr_array[4]+q));
                }

             fclose(fp);

                 //free all the arrays for each set
                 for(int d=0; d<loop; d++)
             {
                free(ptr_array[d]);
             }

   //used in each calculation
```

```
            free(global_array);

        }

        //now were finished we can give back to the heap
        for(int count = 0; count < how_many_times; count++)
        {
            free(input_set_array[count]);
        }

    }
```

Appendix C

Transputer implementation

This appendix gives the source code for the dual axis implementation of the fuzzy PD algortihm.

C.1 Configuration file (dual.cfg)

```
!
!dual.cfg configuration for the TMB008 for dual axis control
! The tasks are named as follows:
! root(slot0) : filter, ip1
! slot1 : output
! slot2 : ip2
!

processor host
processor root
processor slot1
processor slot2

wire ? host[0] root[0]        !anonymous wire connecting PC to transputer
wire ? root[2] slot1[1]       !connect slot0 to slot1 2->1
wire ? slot1[2] slot2[1]      ! connect slot1 to slot2 2->1

!
! Task declarations indicating channel I/O ports and memory requirements
!

task afserver ins=1 outs=1
task filter ins=2 outs=2 data=10K
task ip1 ins=7 outs=7  !leave 0 unconnected!
task ipslv1 ins=2 outs=2 data=100k
task ipslv2 ins=2 outs=2 data=100k
task ipslv3 ins=2 outs=2 data=100k
task out1 ins=3 outs=3 data=200k
task out2 ins=3 outs=3 data=200k
task ip2 ins=7 outs=7
task ip2slv1 ins=2 outs=2 data=100k
task ip2slv2 ins=2 outs=2 data=100k
task ip2slv3 ins=2 outs=2 data=100k


!
! Assign software tasks to physical processors
!

place filter root
place afserver host
place ip1 root
place ipslv1 root
place ipslv2 root
place ipslv3 root
place out1 slot1
place out2 slot1
place ip2 slot2
place ip2slv1 slot2
place ip2slv2 slot2
place ip2slv3 slot2
```

```
!
! Set up the connections between the tasks.
!


bind output ip1[2]  value=&8000000C  ! use output address 3 slot0
bind input  ip1[2]  value=&8000001C  ! <2 is used for a link to the next slot>!

bind output ip2[1]  value=&8000000C  ! use output address 3
bind input  ip2[1]  value=&8000001C  ! <2 is used for a link to the next slot>!

bind output  out1[1] value=&80000000 !output address 0
bind input out1[1]  value=&80000010  !slot1

bind output out2[1]value=&8000000C   !output address 3
bind input out2[1] value=&8000001C   !slot1

!
! bind !! still need to make connections to edge via c4 link switch
!

connect ? afserver[0] filter[0]
connect ? filter[0] afserver[0]

connect ? filter[1] ip1[1]
connect ? ip1[1] filter[1]

connect ? ip1[4] ipslv1[1]
connect ? ipslv1[1]  ip1[4]

connect ? ip1[5] ipslv2[1]
connect ? ipslv2[1] ip1[5]

connect ? ip1[6] ipslv3[1]
connect ? ipslv3[1] ip1[6]

connect ? ip1[3] out1[0]
connect ? out1[0] ip1[3]

!out1[1] used to connect to ip1[2] these are physical links.

connect ? out1[2] out2[2]
connect ? out2[2] out1[2]


connect ? out2[0] ip2[0]
connect ? ip2[0] out2[0]

connect ? ip2[2] ip2slv1[0]
connect ? ip2slv1[0] ip2[2]

connect ? ip2[3] ip2slv2[0]
connect ? ip2slv2[0] ip2[3]

connect ? ip2[4] ip2slv3[0]
connect ? ip2slv3[0] ip2[4]
```

C.2 Header file (declare.h)

```
/*declare.h */

#include <chan.h>
#include <ctype.h>
#include <stdlib.h>

typedef struct{
                float c[4];
              }mess_in;

typedef struct{
                short a[50];
              }arr_no;
```

```
typedef struct{
                float a;
              }fvar;
```

C.3 Header file (smult.h)

```
#define PL      12
#define PM      8
#define PS      4

/* function for scaled implication */
int SMULT_AND_MIN(fvar *err,fvar *edot, arr_no *array1, arr_no *array2)
{
  short dummy=0,av=0,avd=0;
  short min1=0, min2=0;

   av = (short)((err->a*5)+24);
   avd = (short)((edot->a*12.5)+24);

   if(av>49) av = 49;

   if (av<0) av = 0;

   if (avd>49) av = 49;

   if (avd<0) av = 0;

   min1 = array1->a[av];
   min2 = array2->a[avd];

   if (min1 <= min2 )
          min1 = min1;
       else
          min1 = min2;

return(min1);

}
```

C.4 Header file (lrerr5.h)

```
arr_no NLE =
{
100,100,100,100,100,100,99,83,66,49,
33,16,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0
};

arr_no NME =
{
0,0,0,0,0,0,0,16,33,50,
66,83,99,85,71,57,42,28,14,0,
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0
};

arr_no NSE =
{
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,14,28,42,57,71,85,99,
83,66,49,33,16,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0
};

arr_no ZE =
{
```

```
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,16,
33,50,66,83,99,83,66,50,33,16,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
};

arr_no PSE =
{
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,16,33,49,66,
83,99,85,71,57,42,28,14,0,0,
0,0,0,0,0,0,0,0,0,0
};

arr_no PME =
{
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,16,33,49,66,
0,14,28,42,57,71,85,99,83,66,
50,33,16,0,0,0,0,0,0,0
};

arr_no PLE =
{
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,16,33,
49,66,83,99,100,100,100,100,100,100
};
```

C.5 Header file (lrder.h)

```
arr_no NLED=
{
100,100,100,100,100,100,100,100,100,100,
99,79,59,39,19,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
};

arr_no NMED=
{
0,0,0,0,0,0,0,0,0,0,
0,20,40,60,80,100,100,100,100,100,
99,79,59,40,20,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
};

arr_no ZED =
{
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,20,40,59,79,99,80,60,40,20,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
};

arr_no PMED =
{
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,19,39,59,79,
99,100,100,100,100,100,80,60,40,20,
0,0,0,0,0,0,0,0,0,0
};

arr_no PLED = {0,0,0,0,0,0,0,0,0,0,
```

```
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,19,39,59,79,99,
100,100,100,100,100,100,100,100,100,100};
```

C.6 IP1.c

```
/*****************************************************************
* ip1.c  file I/O for passing  structures                        *
* this is the control transputer which deals with i/o and the    *
* calculation of the control signal, coversion of the voltage    *
* to an integer and the integer into 2 bytes format for the      *
* adc and dac conversion is also done here                       *
*****************************************************************/

#include <declare.h>

#undef _DEBUG__
#undef _TIMER__

#ifdef _TIMER__
#include <timer.h>
#endif

#ifdef _DEBUG__
#include <stdio.h>
#include <string.h>
#endif

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
{
int min_sum1, min_sum2, min_sum3;
int op_sum1, op_sum2, op_sum3;
fvar op_total,min_total;
char adcno=0;                    /*adc and dac channel selection numbers*/
short temp_in;
fvar voltout,kd,deriv;
static fvar old_error;
static int sample_skip_count;
short i,j,count;

#ifdef _DEBUG__
FILE *fp;
#endif

#ifdef _TIMER__
float start,end;
static int timer_count;
timer_count=0;
#endif

deriv.a=0.0;
voltout.a=0.0;
old_error.a=0.0;
temp_in=0;
sample_skip_count = 0;

#ifdef _DEBUG__
fp = fopen("data.txt", "w");
i = 0;
#endif

#ifdef _TIMER__
       start = timer_now();
#endif


for(;;)
{
```

```
#ifdef _TIMER__
       timer_count++;
#endif

       chan_out_byte(adcno, out_ports[2]);
       chan_in_message(2, &temp_in, in_ports[2]);

       voltout.a= (float)((temp_in - 2048)*10)>>11;
       voltout.a= (0.2*voltout.a) + (0.8*old_error.a);
       deriv.a = (voltout.a - old_error.a)/2;

#ifdef _TIMER__
         if(timer_count  == 10000)
        {
                  end = timer_now();
                  printf("the input sample time was %f secs\n",\
        (((end-start)*0.000064)/10000));
         }
#endif

  chan_out_message(sizeof(fvar), &voltout.a, out_ports[4]);
  chan_out_message(sizeof(fvar), &deriv.a, out_ports[4]);

  chan_out_message(sizeof(fvar), &voltout.a, out_ports[5]);
  chan_out_message(sizeof(fvar), &deriv.a, out_ports[5]);

  chan_out_message(sizeof(fvar), &voltout.a, out_ports[6]);
  chan_out_message(sizeof(fvar), &deriv.a, out_ports[6]);

  chan_in_message(sizeof(int), &min_sum3, in_ports[6]);
  chan_in_message(sizeof(int), &op_sum3, in_ports[6]);

  chan_in_message(sizeof(int), &min_sum1, in_ports[4]);
  chan_in_message(sizeof(int), &op_sum1, in_ports[4]);

  chan_in_message(sizeof(int), &min_sum2, in_ports[5]);
  chan_in_message(sizeof(int), &op_sum2, in_ports[5]);

  min_total.a = (float)(min_sum1 + min_sum2 + min_sum3);
  op_total.a = (float)(op_sum1 + op_sum2 + op_sum3);

   if((op_total.a == 0) || (min_total.a == 0))
         kd.a = 4;
   else
         kd.a = ((op_total.a)/(min_total.a));

   chan_out_message(sizeof(fvar), &kd.a, out_ports[3]);
   chan_out_message(sizeof(fvar), &deriv.a, out_ports[3]);
   chan_out_message(sizeof(fvar), &voltout.a, out_ports[3]);

 #ifdef _TIMER__
       if(timer_count == 10000)
      {
          end = timer_now();
          printf("the control sample time was %f secs\n",\
                (((end-start)*0.000064)/10000));
      }
 #endif

    if(sample_skip_count == 79)
    {
         old_error.a = voltout.a;
         sample_skip_count = 0;
    }
    else
    {
         old_error.a = old_error.a;
    }
        sample_skip_count++;

  #ifdef _DEBUG__
        while(i<1000)
       {
           fprintf(fp,"%f\t %f \n", kd.a, voltout.a);
           i++;
       }
```

```
        printf("%f\t %f\t %f \n", kd.a, voltout.a,deriv.a);
#endif

}     /* end of for(;;)*/
#ifdef _DEBUG__
      fclose(fp);
#endif
}               /* end of main*/
```

C.7 IP2.c

```
/*************************************************************
* ip2.c       file I/O for passing  structures                *
* this is the control transputer which deals with i/o and the *
* calculation of the control signal, coversion of the voltage *
* to an integer and the integer into 2 bytes format for the   *
* adc and dac conversion is also done here                    *
*************************************************************/

#include <declare.h>
#include <timer.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
{
int min_sum1, min_sum2, min_sum3;
int op_sum1, op_sum2, op_sum3;
fvar op_total, min_total;
char adcno=0;       /*adc and dac channel selection numbers*/
short temp_in;
fvar voltout,kd,deriv;
static fvar old_error;
static int sample_skip_count;

deriv.a=0.0;
voltout.a=0.0;
old_error.a=0.0;
temp_in=0;
sample_skip_count = 0;

for(;;)
{
chan_out_byte(adcno, out_ports[1]);
chan_in_message(2, &temp_in, in_ports[1]);

voltout.a= (float)((temp_in - 2048)*10)>>11;
deriv.a = (voltout.a - old_error.a)/2;


   chan_out_message(sizeof(fvar), &voltout.a, out_ports[2]);
   chan_out_message(sizeof(fvar), &deriv.a, out_ports[2]);

   chan_out_message(sizeof(fvar), &voltout.a, out_ports[3]);
   chan_out_message(sizeof(fvar), &deriv.a, out_ports[3]);

   chan_out_message(sizeof(fvar), &voltout.a, out_ports[4]);
   chan_out_message(sizeof(fvar), &deriv.a, out_ports[4]);

   chan_in_message(sizeof(int), &min_sum3, in_ports[4]);
   chan_in_message(sizeof(int), &op_sum3, in_ports[4]);

   chan_in_message(sizeof(int), &min_sum1, in_ports[2]);
   chan_in_message(sizeof(int), &op_sum1, in_ports[2]);

   chan_in_message(sizeof(int), &min_sum2, in_ports[3]);
   chan_in_message(sizeof(int), &op_sum2, in_ports[3]);


   min_total.a = (float)(min_sum1 + min_sum2 + min_sum3);
   op_total.a = (float)(op_sum1 + op_sum2 + op_sum3);
```

```
    if((op_total.a == 0) ||(min_total.a == 0))
       kd.a = 4;
    else
       kd.a = ((op_total.a)/(min_total.a));


  chan_out_message(sizeof(fvar), &kd.a, out_ports[0]);
  chan_out_message(sizeof(fvar), &deriv.a, out_ports[0]);
  chan_out_message(sizeof(fvar), &voltout.a, out_ports[0]);

/*old card is slightly slower 20 KHz*/
   if(sample_skip_count == 19)
   {
        old_error.a = voltout.a;
        sample_skip_count = 0;
   }
   else
   {
        old_error.a = old_error.a;
   }

   sample_skip_count++;


 }              /* end of for */

 }              /* end of main*/
```

C.8 IPslave1.c

```
/* ipslv1.c        standalone processing task; communicates with master.c */

#include <declare.h>
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];

{
int min1, min2, min3, min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;
derivative.a=0.0;


 for(;;)
 {
     chan_in_message(sizeof(fvar),&err.a,in_ports[1]);
     chan_in_message(sizeof(fvar),&edot.a,in_ports[1]);


         derivative.a=edot.a;

             min1 = SMULT_AND_MIN(&err, &derivative, &PSE, &NMED);
             op1 = min1 * PM;

             min2 = SMULT_AND_MIN(&err, &derivative, &ZE, &NLED);
             op2 = min2 * PS;

             min3 = SMULT_AND_MIN(&err, &derivative, &ZE, &NMED);
             op3 = min3 * PS;

             min4 = SMULT_AND_MIN(&err, &derivative, &ZE, &PMED);
             op4 = min4 * PS;

            op_sum = op1 + op2 + op3 + op4;
            min_sum = min1 +  min2 + min3  + min4;
```

```
    chan_out_message(sizeof(int),  &min_sum, out_ports[1]);
    chan_out_message(sizeof(int),  &op_sum, out_ports[1]);

}
}/* end of main*/
```

C.9 IPslave2.c

```
/* slave2.c        standalone processing task; communicates with master.c */

#include <declare.h>       /* contains type declarations for the stucture */
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
{

int  min1, min2, min3, min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;
derivative.a=0.0;

for(;;)
{

    chan_in_message(sizeof(fvar),&err.a,in_ports[1]);
    chan_in_message(sizeof(fvar),&edot.a,in_ports[1]);

    derivative.a=edot.a;

            min1 = SMULT_AND_MIN(&err, &derivative, &PLE, &NLED);
            op1 = min1 * PM;

            min2 = SMULT_AND_MIN(&err, &derivative, &PLE, &NMED);
            op2 = min2 * PM;

            min3 = SMULT_AND_MIN(&err, &derivative, &PME, &NLED);
            op3 = min3 * PL;

            min4 = SMULT_AND_MIN(&err, &derivative, &PSE, &NLED);
            op4 = min4 * PM;

            op_sum = op1 + op2 + op3 + op4;
            min_sum = min1 +  min2 + min3  + min4;

    chan_out_message(sizeof(int),  &min_sum, out_ports[1] );
    chan_out_message(sizeof(int),  &op_sum, out_ports[1] );

}
}/* end of main*/
```

C.10 IPslave3.c

```
/* ipslv3.c        standalone processing task; communicates with master.c */

#include <declare.h>       /* contains type declarations for the stucture*/
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
```

```
{

int min1, min2, min3,min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;

derivative.a=0.0;

for(;;)
{

   chan_in_message(sizeof(fvar),&err.a,in_ports[1]);
   chan_in_message(sizeof(fvar),&edot.a,in_ports[1]);
derivative.a=edot.a;

            min1 = SMULT_AND_MIN(&err, &derivative, &NSE, &PMED);
            op1 = min1 * PM;

            min2 = SMULT_AND_MIN(&err, &derivative, &NME, &PMED);
            op2 = min2 * PL;

            min3 = SMULT_AND_MIN(&err, &derivative, &NME, &PLED);
            op3 = min3 * PL;

            min4 = SMULT_AND_MIN(&err, &derivative, &NLE, &PLED);
            op4 = min4 * PL;

            op_sum = op1 + op2 + op3 + op4;
            min_sum = min1 +  min2 + min3  + min4;

   chan_out_message(sizeof(int),  &min_sum, out_ports[1] );
   chan_out_message(sizeof(int),  &op_sum, out_ports[1] );


 }
 }/* end of main*/
```

C.11 IP2slave1.c

```
/* ip2slv1.c        standalone processing task; communicates with master.c */

#include <declare.h>           /* contains type declarations for the stucture*/
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>


main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];

{
int min1, min2, min3, min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;

derivative.a=0.0;

for(;;)
{
   chan_in_message(sizeof(fvar),&err.a,in_ports[0]);
   chan_in_message(sizeof(fvar),&edot.a,in_ports[0]);

   derivative.a=edot.a;

            min1 = SMULT_AND_MIN(&err, &derivative, &PSE, &NMED);
            op1 = min1 * PL;

            min2 = SMULT_AND_MIN(&err, &derivative, &ZE, &NLED);
            op2 = min2 * PM;
```

```
            min3 = SMULT_AND_MIN(&err, &derivative, &ZE, &NMED);
            op3 = min3 * PS;

            min4 = SMULT_AND_MIN(&err, &derivative, &ZE, &PMED);
            op4 = min4 * PS;

            op_sum = op1 + op2 + op3 + op4;
            min_sum = min1 +  min2 + min3  + min4;

   chan_out_message(sizeof(int),  &min_sum, out_ports[0] );
   chan_out_message(sizeof(int),  &op_sum, out_ports[0] );

}
}/* end of main*/
```

C.12 IP2slave2.c

```
/* slave2.c        standalone processing task; communicates with master.c */

#include <declare.h>          /* contains type declarations for the stucture*/
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];

{
int min1, min2, min3, min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;
derivative.a=0.0;

for(;;)
{

   chan_in_message(sizeof(fvar),&err.a,in_ports[0]);
   chan_in_message(sizeof(fvar),&edot.a,in_ports[0]);

   derivative.a=edot.a;

            min1 = SMULT_AND_MIN(&err, &derivative, &PLE, &NLED);
            op1 = min1 * PS;

            min2 = SMULT_AND_MIN(&err, &derivative, &PLE, &NMED);
            op2 = min2 * PS;

            min3 = SMULT_AND_MIN(&err, &derivative, &PME, &NLED);
            op3 = min3 * PM;

            min4 = SMULT_AND_MIN(&err, &derivative, &PSE, &NLED);
            op4 = min4 * PL;

            op_sum = op1 + op2 + op3 + op4;
            min_sum = min1 +  min2 + min3  + min4;

    chan_out_message(sizeof(int),  &min_sum, out_ports[0] );
    chan_out_message(sizeof(int),  &op_sum, out_ports[0] );

 }
 }/* end of main*/
```

C.13 IP2slave3.c

```
 /* ipslv3.c        standalone processing task; communicates with master.c */
```

```
fvar voltout,kd,deriv,volt_signal;
static float signal;
int start,end;
deriv.a=0.0;

for(;;)
  {

     chan_in_message(sizeof(fvar), &kd.a, in_ports[0]);
     chan_in_message(sizeof(fvar), &deriv.a, in_ports[0]);
     chan_in_message(sizeof(fvar), &voltout.a, in_ports[0]);

     signal=(float)(-1*(0.42*voltout.a + kd.a*deriv.a));

     signal = ((signal<<11)/10)+2048;

     temp_out=(short)(signal);

     reset1=16776960;
     reset0=16777215;

     chan_out_message(4,&reset0,out_ports[1]);
     chan_out_message(4,&reset1,out_ports[1]);
     chan_out_message(2,&temp_out,out_ports[1]);
     chan_out_byte(1,out_ports[1]);
  }

}
```

C.15 out2.c

```
/*****************************************************************
* out2.c file I/O for passing structures example                 *
* this is the new card - used for up and down movement           *
* this is the control transputer which deals with i/o and the    *
* calculation of the control signal, coversion of the voltage    *
* to an integer and the integer into 2 bytes format for the      *
* adc and dac conversion is also done here                       *
* this output is used for the side_to_side movement of the arm   *
* it is bound to edge 2 by the c400 link switch                  *
*****************************************************************/


#include <declare.h>
#include <timer.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
{

arr_no min;
int reset0,reset1;
char dacno=1;                              /*adc and dac channel selection numbers*/
char outhi,outlo;                           /*high and low bytes for i/o          */
short error_shrt,temp_in,temp_out,volt_temp;
fvar voltout,kd,kp,deriv,volt_signal;
static float signal;
int start,end;
deriv.a=0.0;
temp_in=0;


    for(;;)
    {
        chan_in_message(sizeof(fvar), &kd.a, in_ports[0]);
        chan_in_message(sizeof(fvar), &deriv.a, in_ports[0]);
        chan_in_message(sizeof(fvar), &voltout.a, in_ports[0]);

    signal=(float)(-1*(0.42*voltout.a + kd.a*deriv.a));

        signal = ((signal<<11)/10)+2048;
```

```
#include <declare.h>
#include <lrder.h>
#include <lrerr5.h>
#include <smult.h>


main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];

{
int min1, min2, min3, min4, min_sum;
int op1, op2, op3, op4, op_sum;
fvar err, edot;
static fvar derivative;
derivative.a=0.0;

for(;;)
{

    chan_in_message(sizeof(fvar),&err.a,in_ports[0]);
    chan_in_message(sizeof(fvar),&edot.a,in_ports[0]);

    derivative.a=edot.a;

             min1 = SMULT_AND_MIN(&err, &derivative, &NSE, &PMED);
             op1 = min1 * PM;

             min2 = SMULT_AND_MIN(&err, &derivative, &NME, &PMED);
             op2 = min2 * PL;

             min3 = SMULT_AND_MIN(&err, &derivative, &NME, &PLED);
             op3 = min3 * PL;

             min4 = SMULT_AND_MIN(&err, &derivative, &NLE, &PLED);
             op4 = min4 * PL;

             op_sum = op1 + op2 + op3 + op4;
             min_sum = min1 +  min2 + min3  + min4;

    chan_out_message(sizeof(int),  &min_sum, out_ports[0] );
    chan_out_message(sizeof(int),  &op_sum, out_ports[0] );

 }
 }/* end of main*/
```

C.14 out1.c

```
/*************************************************************
* out1.c file I/O for passing structures                     *
* movement in the horizontal plane side-to-side              *
* old cards !!!!                                             *
* this is the control transputer which deals with i/o and the *
* calculation of the control signal, coversion of the voltage *
* to an integer and the integer into 2 bytes format for the  *
* adc and dac conversion is also done here                   *
*************************************************************/


#include <declare.h>

main(argc, argv, envp, in_ports, ins, out_ports, outs)
int argc, ins, outs;
char *argv[], *envp[];
CHAN *in_ports[], *out_ports[];
{

int reset0,reset1;
char dacno=1;                   /*adc and dac channel selection numbers*/
char outhi,outlo;                /*high and low bytes for i/o          */
short temp_out,volt_temp;
```

```
        temp_out=(short)(signal);


        reset1=16776960;
        reset0=16777215;

       chan_out_message(4,&reset0,out_ports[1]);
       chan_out_message(4,&reset1,out_ports[1]);
       chan_out_message(2,&temp_out,out_ports[1]);
       chan_out_byte(dacno,out_ports[1]);
    }

}
```

Appendix D

Transputer data acquisition datasheets

Datasheet from:-

Industrial Development Bangor (UCNW) Ltd.,
Dean Street,
Bangor,
Gwynedd,
Wales,
LL57 1UT.

SPECIFICATION

Number of channels:	o 8
Sample rate:	o 16μs
ADC:	o Analog Devices AD7572JN, 5μs conversion o ± 10.0V bipolar input o sample & hold on analog input o 12 bit resolution o ± 1 l.s.b. linearity
Transputer Interface:	o IMS C011 link adaptor with 5MHz clock o 10/20 Mbit/s selectable link speed o 10μs communication time for 12 bit data
Typical power requirements:	o 5V at 50mA o +15V at 7mA o -15V at 12mA
Format:	o Single Eurocard with DIN41612 connector (64 way a/b plug).

Figure D.1: Specification for Analogue to Digital Card - IDB 534/1.

SPECIFICATION

Number of channels:	o 4 or 8. The basic version has 4 channels; upgrading to 8 requires insertion of 1 more dac chip
Sample rate:	o 18μs (with 10Mbit/s link rate option)
DAC:	o Analog Devices AD390JD o maximum 8μs settling to ± 1/2 lsb o ± 10V bipolar output at 5mA maximum o 12 bit resolution and accuracy o ± half lsb linearity
Transputer Interface:	o IMS C011 link adaptor with 5MHz clock o 10 or 20 Mbit/s link rate, selectable o 10μs communication for 12 bit data at 10Mbit/s
Typical power requirements:	o 5V at 63mA o +15V at 20mA o -15V at 72mA (4 channel option)
Format:	o Single Eurocard with DIN41612 connector (64 way a/b plug).

Figure D.2: Specification for Digital to Analogue Card - IDB 534/2.

Appendix E

Harmonic Drive datasheet

RHS/RFS Series DC-Servo-Actuators

Main Specifications

Rating:	Continuous	Construction:	totally closed	Vibration:	2.5G (5 ~ 400Hz)
Exciting method:	Permanent magnet	Ambient temperature:	-10 ~ +40°C	Color:	NTK89-163 (Gray-blue metallic)
Motor Insulation:	Class F	Storage temperature:	-20 ~ +60°C	Output:	Shaft (RHS series)
Insulation voltage:	AC 1000V, one minute	Ambient humidity:	35 ~ 80%		Flange (RFS Series)
Insulation resistance:	100MΩ (DC500V Megger)		(Non-condensing)	Lubrication:	Grease HC-1

Technical Data

Item	Actuator	RHS-20, RFS-20				RHS-25, RFS-25				RHS-32, RFS-32			
		6007	3007	6012	3012	6012	3012	6018	3018	6018	3018	6030	3030
	Applied Control Unit	HS-250-3 HS-350-3 HS-450 G/R-3				HS-250-6 HS-350-6 HS-450 G/R-6				HS-250-9 HS-350-9 HS-450 G/R-9			
Rated output power [1]	W	74	74	123	111	123	123	185	185	185	185	308	308
Rated voltage [1]	V	75	75	75	75	75	75	75	75	75	75	85	85
Rated current [1]	A	1.9	1.9	2.9	2.7	3.1	3.1	3.9	3.9	4.1	4.1	5.4	5.4
Rated output torque [1] T_N	Nm	12	24	20	36	20	40	30	60	30	60	50	100
Rated output speed [1] n_N	rpm	60	30	60	30	60	30	60	30	60	30	60	30
Continuous stall torque [1][2]	Nm	14	28	24	43	24	48	36	72	36	72	60	120
Peak current [1][2]	A	6.2	4.8	6.4	5.0	10.7	8.8	10.0	8.3	20.6	16.3	18.2	14.4
Repeated peak torque [1][2] T_R	Nm	57	84	57	84	100	160	100	160	220	340	220	340
Maximum output speed [1]	rpm	80	40	80	40	80	40	80	40	80	40	80	40
Torque constant	Nm/A	10.5	21.0	10.5	21.0	10.5	21.0	11.5	22.9	11.5	22.9	13.3	26.6
Voltage constant (B.E.M.F.)	V/rpm	1.08	2.15	1.08	2.15	1.08	2.15	1.18	2.35	1.18	2.35	1.37	2.74
Moment of inertia [3]	kgm²	0.29	1.2	0.47	1.9	0.53	2.1	1.1	4.5	1.4	5.8	3.1	12.0
Mechanical time constant	msec	9.2	9.2	5.1	5.1	5.7	5.7	5.2	5.2	6.8	6.8	7.0	7.0
Rated power rate [1]	kW/s	0.48	0.48	0.83	0.67	0.74	0.74	0.79	0.79	0.61	0.61	0.80	0.80
Thermal time constant [1]	min	19	19	21	21	23	23	24	24	25	25	29	29
Thermal resistance [1]	°C/W	1.14	1.14	0.99	0.99	0.93	0.93	0.76	0.76	0.71	0.71	0.51	0.51
Gear ratio		50	100	50	100	50	100	50	100	50	100	50	100
Maximum radial load	N	RHS:1400		RFS:2000		RHS:3000		RFS:2500		RHS:4500		RFS:4000	
Maximum thrust load	N	RHS:1400		RFS:900		RHS:3000		RFS:1100		RHS:4500		RFS:1600	
Motor rated output [1]	(W)	(120)	(120)	(200)	(200)	(200)	(200)	(300)	(300)	(300)	(300)	(500)	(500)
Motor rated speed [1]	(rpm)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)	(3000)
Armature resistance	Ω	3.4	3.4	1.2	1.2	1.2	1.2	0.60	0.60	0.60	0.60	0.40	0.40
Armature inductance	mH	2.7	2.7	1.1	1.1	1.1	1.1	0.92	0.92	0.92	0.92	0.84	0.84
Electrical time constant	msec	0.81	0.81	0.93	0.93	0.93	0.93	1.5	1.5	1.5	1.5	2.1	2.1
Starting current	A	0.5	0.35	0.5	0.35	0.7	0.45	0.7	0.45	1.1	0.75	1.1	0.75
No load running current [4]	A	0.8	0.8	1.0	1.0	1.2	1.2	1.3	1.3	1.5	1.5	1.6	1.6

Table 13

Please Note:

[1] The values are for saturated actuator temperature. Other values (not marked with [1]) are for actuator temperature of 20°C.
[2] The values given represent an upper limit and actual load values should be lower.
[3] The tabulated value is the moment of inertia referred to the output shaft resulting from the sum of the motor inertia and Harmonic Drive inertia.
[4] Values are for rated output speed.

Additional information

- Actuator specifications show output characteristics including gear efficiency.
- All specifications are applicable for actuators mounted on aluminium heat sink of the following sizes:
  RHS/RFS-20: 250 x 250 x 12 mm,
  RHS/RFS-25 and
  RHS/RFS-32:300 x 300 x 15 mm.

Figure E.1: Technical data for the Harmonic Drive DC servo

Appendix F

Labview data acquisition

F.1 User interface

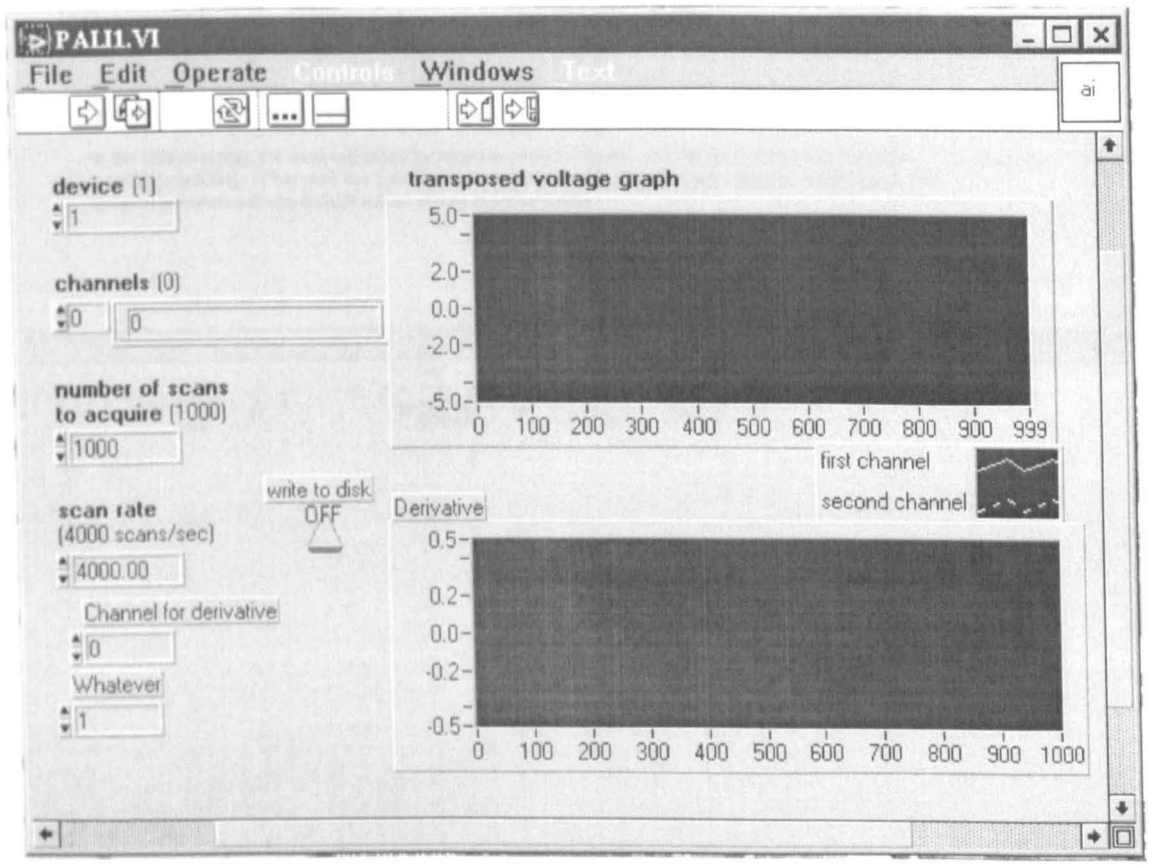

Figure F.1: Labview user Interface

F.2 Design

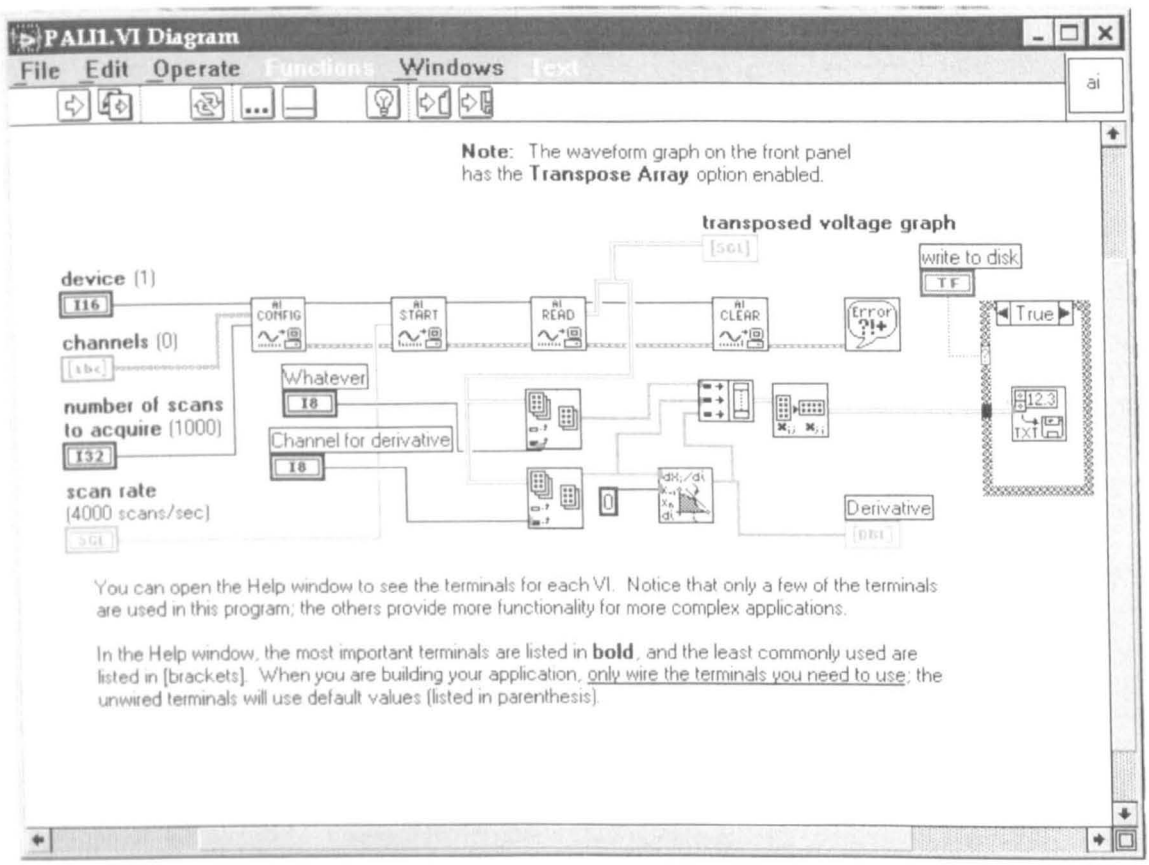

Figure F.2: Labview design